# TEK ÜRETİCİLİ ÇOK MÜŞTERİLİ BİR SİSTEMDE TAŞIMA PLANLANMASI

**MEHMET SERKAN TOKGÖZ**

**YÜKSEK LİSANS TEZİ**

**ENDÜSTRİ MÜHENDİSLİĞİ**

**TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ**

**FEN BİLİMLERİ ENSTİTÜSÜ**

**ARALIK 2015**

**ANKARA**

Fen bilimleri enstitü onayı

_____

Prof. Dr. Osman EROĞUL
Müdür

Bu tezin Yüksek Lisans derecesinin tüm gereksinimlerini sağladığını onaylarım.

_____

Prof. Dr. Tahir HANALİOĞLU
Anabilim Dalı Başkanı

Mehmet Serkan TOKGÖZ tarafından hazırlanan TEK ÜRETİCİLİ ÇOK MÜŞTERİLİ BİR SİSTEMDE TAŞIMA PLANLANMASI adlı bu tezin Yüksek Lisans tezi olarak uygun olduğunu onaylarım.

_____

Doç. Dr. Kadir ERTOĞRAL
Tez Danışmanı

Tez Jüri Üyeleri

Başkan : Yrd. Doç. Dr. Ramazan ŞAHİN _____

Üye : Doç. Dr. Kadir ERTOĞRAL _____

Üye : Yrd. Doç. Dr. Eda YÜCEL _____

## TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada orijinal olmayan her türlü kaynağa eksiksiz atıf yapıldığını bildiririm.

Mehmet Serkan Tokgöz

| | | |
|---|---|---|
| **Üniversite** | **:** | TOBB Ekonomi ve Teknoloji Üniversitesi |
| **Enstitü** | **:** | Fen Bilimleri |
| **Anabilim Dalı** | **:** | Endüstri Mühendisliği |
| **Tez Danışmanı** | **:** | Doç. Dr. Kadir ERTOĞRAL |
| **Tez Türü ve Tarihi** | **:** | Yüksek Lisans – ARALIK 2015 |

**Mehmet Serkan TOKGÖZ**

# TEK ÜRETİCİLİ ÇOK MÜŞTERİLİ BİR SİSTEMDE TAŞIMA PLANLANMASI

## ÖZET

Bu çalışmada birden fazla bölgede coğrafi olarak dağılmış bulunan bayiler ya da müşterilere ulaştırılacak ürünlerin taşıma planlaması ele alınmaktadır. Üretici ürünlerin dağıtımını lojistik firmalarına yaptırmaktadır. Dağıtımda ya firmalardan kamyon kiralanmakta veya parsiyel taşıma şeklinde ürünler müşterilere parsiyel kargo taşıyıcı firmalar kanalıyla ulaştırılmaktadır. Müşterilerin farklı ürünlere olan talepleri belirli bir planlama periyodu boyunca bilinmektedir. Müşterilere talepleri, talep edilen periyoda kadar taşınmak zorundadır. Müşteriler belirli bölgelere ayrılmışlardır. Ürünlerin taşındığı kamyonların tiplerine göre kapasiteleri vardır ve bu kapasiteler bilinmektedir. Taşıma planlaması yapılırken bir kamyon tarafından ziyaret edilen müşteri başına maliyet, aracın kullanılmasına ve gittiği bölgeye bağlı sabit maliyet ve parsiyel taşıma maliyeti olarak taşınan ürünlerin toplam maliyeti hesaplanmaktadır. Problem için literatürdeki sabitle ve optimize et sezgisel metodu uyarlanarak kullanılmıştır. Sezgisel metodun matematiksel modele göre performansı rassal olarak üretilen problem setleri üzerinden değerlendirilmiştir.

**Anahtar Kelimeler:** Taşıma Planlaması, Matematiksel Modelleme, Sabitle & Optimize et Algoritması

| | | |
|---|---|---|
| **University** | **:** | TOBB University of Economics and Technology |
| **Enstitute** | **:** | Institute of Natural and Applied Science |
| **Science Programme** | **:** | Industrial Engineering |
| **Supervisor** | **:** | Assoc. Prof. Kadir ERTOĞRAL |
| **Degree Awarded & Date** | **:** | M.Sc. – DECEMBER 2015 |

**Mehmet Serkan TOKGÖZ**

**A FIX AND OPTIMIZE HEURISTIC FOR TRANSPORTATION PLANNING IN A SINGLE PRODUCER MULTI BUYER PROBLEM**

## ABSTRACT

We address the transportation planning problem for a producer who supplies products to geographically dispersed customers with known demands for a finite planning horizon. The problem originated from a real case which involves a producer of dry pulses, who serves entire Turkey from two production plants. The producer uses services of logistic companies in transporting goods in the form of both renting trucks from a company and using parcel transportation service from a carrier. The decisions that the producer has to make in each period are which customers' demand to load to which trucks, and for which customers' demand to use parcel carrier service. Total cost of transportation is the summation of the fixed renting cost of sending a truck to a zone, a cost depending on the total number of customers visited, and parcel cost. We introduced a mathematical programming model of this problem for the first time in the literature, and suggested a fix and optimize type heuristic solution procedure. We showed the effectiveness of the suggested heuristic on a set of random problem.

**Keywords:** Transportation planning, Mathematical Modelling, Fix&Optimize

## TEŞEKKÜR

Çalışmam boyunca yardım etmekten ve bilgisini paylaşmaktan çekinmeyen sayın danışman hocam Doç. Dr. Kadir ERTOĞRAL'a teşekkürü borç bilirim. Manevi destekleri için önce kendi aileme daha sonra tüm TOBB ETU ailesine teşekkürü borç bilirim. Ayrıca yüksek lisans öğrenimim boyunca her ay bana burs sağlayan üniversitem TOBB ETU'ye teşekkür ederim.

# İÇİNDEKİLER

# ŞEKİL LİSTESİ

## TABLO LİSTESİ

## KISALTMALAR

| Kısaltmalar | Açıklama |
|---|---|
| **ÇP** | Çok periyot bazlı yaklaşım |
| **TP** | Tek periyot bazlı yaklaşım |
| **İPKP** | İki periyotlu kayan pencere yöntemi |
| **AP** | Araç periyot bazlı yaklaşım |
| **BAÖP** | Büyük araç öncelikli & periyot bazlı yaklaşım |
| **SOE 1** | Sabitle ve optimize et algoritması aşama 1 |
| **SOE 2** | Sabitle ve optimize et algoritması aşama 2 |
| **AOF** | Alt sınır optimal farkı |

# 1.GİRİŞ

Günümüzde en önemli konular "küreselleşme ve değişim" olarak belirlenmektedir. Çağımızda yıllar geçtikçe yaşanan değişim ve gelişme, önceki dönemlerle karşılaştırılamayacak kadar hızlıdır. Yeni teknolojinin bulunması, yeni üretilen icatlar, bilgi işlem sektöründeki gelişmeler, ekonomide yaşanan süreçlerin çağımızdaki hızı nitelendirilemeyecek kadar fazladır. Bu nedenle gün geçtikçe küreselleşen Dünyamızda üretilen ürünlerin pazara ve müşterilere zaman kaybetmeden, hızlı bir şekilde ulaştırılması neredeyse zorunluluk oluşturmaktadır. Ülkeler ve bölgeler arasındaki sınırların azalan önemi sayesinde Dünya küresel bir köye dönüşmektedir. Bu nedenle küreselleşmenin ve değişimin bu denli arttığı çağımızda, mal ve hizmet ticareti önemli bir yer almıştır. Mal ve hizmet ticareti gelişirken, şirketler için ulaştırma faktörü hem maliyetler açısından hem de pazarda daha çabuk yer alma isteği açısından oldukça önemli bir faktör haline gelmiştir. Mesafelerin fazlalığı, zamanın önemi gibi parametrelerin ortaya çıkması ve artan rekabet koşulları şirketleri birden fazla ulaşım yöntemini bütünleşmiş bir şekilde kullanıp en iyi sonucu alma çabasına sürüklemiştir. Gösterilen bu çaba da şirketleri bir taşıma planlaması kararının oluşturulması için ikna etmeye yeterlidir. Taşıma planlaması içerisinde verilen tüm stratejik kararlar, ulaştırma operasyonlarının tüm aşamalarını içinde bulunduran sistemin oluşmasına neden olmuştur. Bu sisteme de lojistik adı verilmiştir. Lojistik en genel tanımıyla bir ürünü tedarikçiden nihai tüketicisine ulaştırmak için gerekli tüm faaliyetler olarak tanımlanabilmektedir [10,20].

Genellikle lojistik aşağıdaki konuları yönetmektedir

- Sipariş işleme
- Satın alımlar
- Gelen ulaşım
- Üretim planları ve programları
- Envanter yönetimi

- Dağıtım ve teslimat ulaşımı
- Depo yönetimi
- Malzeme ve dağıtım ihtiyaç planlaması

Özellikle son 20 yılda lojistik kavramı iş dünyasında çok duyulmakta ve önem kazanmaktadır. Ticaret sektöründe lojistiğe bakış açısı daha fazla iş odaklıdır. Lojistik, müşterilerin ihtiyaçlarını karşılamak için hammaddenin etkin maliyetle akışı, depolanması, işlenmesi ile tamamlanmış ürün elde edilmesi ve ilgili bilginin kaynak noktasından tüketim noktasına kadar ulaşmasını planlayan, uygulayan ve kontrol eden işlemdir. Lojistik olmadan; pazarlama, üretim ve uluslararası ticarette başarılı olmak olanaksızdır. Gelişmiş endüstriyel toplumlarda lojistik yeterlilik büyük önem taşımaktadır. Bunun nedeni tüketicilerin, satın aldıkları ürünlerin söz verildiği gibi teslim edilmesini istemesidir[10,20].

Türkiye'deki lojistik şirketleri çoğunlukla dış ticaretin yoğun olduğu sektörlere hizmet vermektedir. Ülkemizde genel olarak nakliyat hizmetleri olarak bilinen lojistik hizmetler, bir ya da daha fazla taşıma türü kullanılarak gerçekleştirilen taşımaya ilişkin konsolidasyon, depolama, elleçleme, paketleme veya dağıtım olduğu kadar; ilave işlemler ve danışmanlık dahil tüm hizmetler ile eşyanın gümrük beyanının yapılması, sigortalanması, kıymetli evrakın hazırlanması ve ödemenin tahsilat gibi işlemleri de kapsayan hizmetler zincirini ifade etmektedir.

Lojistikte belirli maliyetleme yöntemleri bulunmaktadır. Bu maliyetlendirme yöntemleri içsel ve dışsal olarak birbirinden ayrılabilmektedir. Maliyetin büyük bir kısmını ürünlerin taşıması, depolanması, talep tahmini, üretim planlaması, müşteri hizmetleri, stok yönetim, satın salınması, sipariş yönetimi gibi unsurlar oluşturmaktadır. Bu faaliyetlerinin genelinin oluşturduğu bütünlüğe lojistik, maliyetlerin toplamına ise lojistik maliyetleri ismi verilmektedir[10].

İçsel lojistik maliyetler aşağıdaki gibi sıralanmaktadır.

- Her bir taşımacılık moduna göre maliyetler
- Depolama maliyetleri
- Bozulma, hasar ve kayıp maliyetleri

- Geç teslim, ceza, hata, plansızlık, kapasite maliyetleri
- Stok bulundurma maliyeti
- Taşıma ve depolama amaçlı paketleme maliyetleri
- Bilişim ve iletişim maliyetleri
- Elleçleme vd. ürün aktarım maliyetleri
- Elde bulundurmama maliyetleri
- Lojistik yönetim maliyetleri

Lojistikteki dışsal maliyetler ise genellikle aşağıdaki gibi sıralanmaktadır.

- Kaza maliyeti
- Hava kirliliği maliyeti
- Gürültü maliyeti
- Sıkışıklık Kaynaklı Maliyet

Toplam maliyet faktörü bahsedilen içsel ve dışsal maliyetlerin yönetiminin entegre bir şekilde yapıldığı ve bu toplam maliyetin minimizasyonu ile ilgilidir. Birçok firmada lojistiğin dört karar alanı olan nakliye, stok kontrol, depolama ve müşteri hizmetlerinin verimli biçimde kullanılması büyük maliyet faydası sağlamaktadır. Lojistikte bu karar alanları arasında maliyet dengesi önemlidir. Çünkü bir alan beklenmedik şekilde diğer alanı da etkileyebilmektedir. Günümüzde lojistik maliyetlerin toplam ürün maliyetleri içindeki payı yaklaşık %40'lara kadar ulaşmıştır. İşletmeler bahsedilen bu lojistik maliyetleri düşürmek için ciddi arayışlar içine girmektedirler. Bu nedenle bir taşımacılık sisteminde, göndericiler ve hizmet alanlar bir noktada buluşarak kendileri için en uygun ve karlı olanı seçerler. Karar kılınan noktada maliyet, verim ve performans değerleri sistemde bulunan taşıyıcı firmalar, göndericiler ve müşteriler için optimal değere sahip olmalıdır[10,20].

Bu çalışmada problem üretilirken daha önceden bir bakliyat firmasında yaşanan problemden esinlenilmiştir. Bakliyat firması Ankara, Mersin ve Çankırı'da sahip olduğu üç büyük tesisi ile Türkiye'nin öncül üreticilerinden birisi konumundadır. Firmanın hedefleri arasında müşteri memnuniyeti, müşterilere siparişi zamanında ve en düşük maliyet ile ulaştırmak vardır. Bu çizgide çalışmalarını devam ettiren firma,

müşterilere ürünlerini kendi tırları ile, anlaşmalı lojistik firmaları ile veya parsiyel taşımacılık ile ulaştırmaktadır. Lojistik düzeyde sevkiyat listelerini firmanın lojistik departmanında çalışan personel kendi tecrübesine güvenerek rasgele oluşturmaktadır. Sipariş edilen ürünün kamyonlara atanması da tecrübeye dayalı bir şekilde gerçekleştirilmektedir. Sevkiyat listeleri anlık olarak gelen her siparişten sonra değişkenlik gösterebilmektedir. Talep dinamik olarak geldiği için çok uzun süreli sevkiyat planı yapmak oldukça zordur. Her müşterinin siparişlerini karşılama zorunluluğu olduğu için mevcut sistemde araçların siparişleri götürürken tam kapasite kullanılamaması ve taşıma planlarının etkin yapılamaması, siparişleri zamanında göndermek için parsiyel taşımacılığın alternatif taşıma olarak kullanılmasına yol açmaktadır. Bu da katlanılan taşıma maliyetlerinin artmasına neden olmaktadır. Firmanın verilerine göre, parsiyel taşımacılık maliyeti normal lojistik firmasından yararlanılan taşımacılığa oranla neredeyse 4 kat daha maliyetli olabilmektedir. Ayrıca taşımada büyük kapasiteli araçların yüksek doluluk oranıyla kullanılması maliyetleri azaltacaktır. Firma parsiyel maliyeti ve toplam taşıma maliyetini azaltmak için çözüm yöntemleri aramaktadır.

Bu çalışmada bakliyat firmasının probleminden esinlenerek literatürde göremediğimiz yeni bir problem oluşturulmuştur. Problem için matematiksel modeller önerilmiş ve literatürden bilinen sabitle ve optimize et sezgisel metodu uyarlanarak problem üzerinde uygulanmış ve performansı incelenmiştir.

Çalışmanın anlatımına problemin tanımlanması ve daha sonra tanımlanan problem için geliştirilen formülasyon ile başlanacaktır. Daha sonra tanımlanan problem ile ilgili literatürde araştırılan makalelerden bahsedilecektir. Uygulanan çözüm yöntemi bölümünde tanımlanan problem için geliştirilen sabitle ve optimize et sezgiselinin çalışma prensibi ve probleme nasıl uygulandığı anlatılacaktır. Literatürde tam olarak karşılığı bulunmayan problemimiz için örnek veri üretimi ve veri üretirken kullanılan parametrelerin tanımları numerik analiz bölümünde anlatılacaktır. Üretilen bu örnek problemler üzerinde sezgisel metodun performansı ise problem çözümü ve analizi bölümünde anlatılacaktır. Daha sonra sonuçların değerlendirilmesi ve kaynaklar ile çalışmanın anlatımı sonlandırılacaktır.

## 2. PROBLEM TANIMI VE FORMÜLASYON

Bu tez çalışmasında tek üreticili çok müşterili bir sistem için taşıma planlaması yapılmıştır. Problemde B adet bölgeye coğrafik olarak yayılmış M adet müşteriye T adet periyottan oluşan planlama ufkunda talepleri ulaştıran bir üreticimiz vardır. Bir müşteri konumuna göre birden fazla bölgeye ait olabilmektedir. Talepler ulaştırılırken iki seçenek vardır; lojistik hizmet veren şirketlerden V adet kapasiteleri farklı olan araçlar kiralanabilmektedir veya talepler parsiyel olarak taşınabilmektedir. Fakat belirtildiği üzere parsiyel taşımacılık maliyeti kiralanan araç ile taşımaya oranla çok daha yüksek maliyetlidir. K adet farklı ürün bulunmaktadır ve müşterilerin her periyotta her ürün için olan talepleri bilinmektedir. Problem aracın bölgeye gitmesine göre maliyet hesaplayan bölge bazlı model ve bölgede gidilen en uzak müşterinin mesafesine dayalı maliyetini hesaplayan uzaklık bazlı model olmak üzere iki ayrı şekilde incelenmiş ve modellenmiştir. Bu iki tür maliyet yapısı da bileşenleri itibariyle literatürde kullanılmıştır. Problemin başlıca varsayımları aşağıda maddeler halinde sıralanmıştır. Fakat numerik analiz kısmını ve sonuçları karşılaştırmayı firmanın daha sık kullanması sebebiyle çalışma boyunca bölge bazlı model ile ilgilenilmiştir. Bölge bazlı model ve uzaklık bazlı modelin genel yapıları aynıdır ve aşağıdaki şartlar her iki model için de geçerlidir.

- Talepler bölünememektedir. Bu nedenle bir müşterinin aynı periyottaki talebi tek bir araçla taşınabilmektedir.
- Müşteri talepleri önceden taşınabilir ancak belirli bir periyottan önce taşınamaz.
- Her müşterideki talepler her periyot için planlama ufku boyunca bilinmektedir.
- Kiralanan bir araç, aynı periyotta en fazla bir bölgeye gidebilmektedir.
- Araç rotalama ve ilgili maliyetler araç sahibi firmayı ilgilendiren bir konudur ve çalışılan probleme dahil edilmemiştir.
- Periyotlar esinlenilen gerçek hayat probleminde güne denk gelmektedir.

Yukarıdaki varsayımlardan en kısıtlayıcı olan taleplerin bilinmesi varsayımıdır. Literatürde belirli talep altında bile bu problem incelenmemiştir ve bu varsayım altında geliştirilen model problemin analizinde bir ilk adım olacaktır. Bu varsayımlar altında, problemin amacı toplam taşıma maliyetini minimize etmektir. Maliyetler toplamı üç ana faktörden oluşmaktadır. Bu maliyeter aracın her bir müşteride durma veya ziyaret başına maliyeti, hangi aracın hangi bölgeye gönderildiğine göre değişen maliyet ve parsiyel taşımacılık seçeneği ile taşınan kısım ile ilgili maliyettir.

**Problem Parametreleri**

*B= Bölgeler kümesi*

$B_{b=}$ *b bölgesinde bulunan müşterilerin kümesi*

*M= Müşteriler kümesi*

*K= Ürünler kümesi*

*V= Araçlar kümesi*

$d_{jit} = j$ müşterisinin $i$ ürününden $t$ periyodunda talep ettiği miktar

$p_{ij} = j$ müşterisinin $i$ üründen talebini parsiyel taşımanın maliyeti

$C_{vb} = v$ aracını $b$ bölgesine göndermenin maliyeti

$Cap_v = v$ aracının kapasitesi

*Sv = v aracının durma başına maliyeti*

$D_j = j$ müşterisine olan maksimum uzaklık

*w = Müşterinin talebinin kaç periyot erken gönderilebileceğini sınırlayan parametre*

**Karar Değişkenleri**

$$X_{jitvt'} = \begin{cases} 1, eğer\ j\ müşterisinin\ i\ ürününden\ t'periyodundaki\ talebi \\ \quad v\ aracı\ ile\ t\ periyodunda\ taşınıyor\ ise \\ 0, aksi\ taktirde \end{cases}$$

$$y_{vbt} = \begin{cases} 1, v\ aracı\ t\ periyodunda\ b\ bölgesine\ gönderildiyse \\ 0, aksi\ taktirde \end{cases}$$

$$L_{vt} = v\ aracının\ t\ periyodundaki\ yükü$$

$$R_{jit} = j\ müşterisinin\ i\ ürünü\ talebinden\ t\ periyotunda\ parsiyel\ taşınan\ miktar$$

$$dist_{vt} = v\ aracının\ t\ periyodunda\ katettiği\ maksimum\ mesafe$$

## 2.1. Matematiksel Model

### 2.1.1. Bölge Bazlı Model

$$Min\ z = \sum_{v\in V}\sum_{t\in T} C_{vb}y_{vbt} + \sum_{j\in M}\sum_{i\in K}\sum_{t\in T} R_{jit}p_{ij} + \sum_{i\in K}\sum_{t\in T}\sum_{V\in V}\sum_{t'=1}^{T}\sum_{t=t'}^{T}\sum_{j\in M}\sum_{i\in K} X_{jit'vt}S_v$$

$$\sum_{v\in V}\sum_{t'=1}^{t} X_{jit'vt} \le 1 \qquad \forall j \in M;\ \forall i \in K\ ;\ \forall t \in T \qquad (2.1.1.1)$$

$$\sum_{v\in V}\sum_{t'=1}^{t-w} X_{jit'vt} = 0 \qquad \forall j \in M;\ \forall i \in K;\ \forall t \in T \qquad (2.1.1.2)$$

$$L_{vt'} = \sum_{j\in M}\sum_{t=t'}^{T}\sum_{i\in K} X_{jit'vt} d_{jit} \qquad \forall t' \in T;\ \forall v \in V \qquad (2.1.1.3)$$

$$R_{jit} = d_{jit} - \sum_{v\in V}\sum_{t'=1}^{t} X_{jit'vt} d_{jit} \qquad \forall j \in M;\ \forall i \in K;\ \forall t \in T \qquad (2.1.1.4)$$

$$L_{vt} \le \sum_{b\in B} y_{vbt} Cap_v \qquad \forall v \in V;\ \forall t \in T \qquad (2.1.1.5)$$

$$\sum_{t=t'}^{T}\sum_{i\in K}\sum_{j\in B_b} X_{jit'vt} \le y_{vbt'} M \qquad \forall b \in B;\forall v \in V;\ \forall t' \in T \qquad (2.1.1.6)$$

$$\sum_{b\in B} y_{vbt} \le 1 \qquad \forall v \in V;\ \forall t \in T \qquad (2.1.1.7)$$

$$X_{jit'vt} \in \{0,1\} \qquad \forall j \in M;\ \forall i \in K;\ \forall t \in T;\forall v \in V;\ \forall t \in T \qquad (2.1.1.8)$$

$$y_{vbt} \in \{0,1\} \qquad \forall b \in B;\forall v \in V;\ \forall t \in T \qquad (2.1.1.9)$$

$$L_{vt} \ge 0 \qquad \forall v \in V;\ \forall t \in T \qquad (2.1.1.10)$$

$$R_{vit} \ge 0 \qquad \forall v \in V;\ \forall i \in K;\ \forall t \in T \qquad (2.1.1.11)$$

Amaç fonksiyonu aracın bölgeye gitme maliyeti, durma başına maliyeti ve müşteri talebinin parsiyel taşıtılma maliyetini kapsamaktadır. Problemin amacı bu toplam maliyeti minimize etmektedir. Kısıt (2.1.1.1) bir müşteri talebinin aynı periyot içerisinde yalnızca bir araca atanmasını sağlamaktadır. Kısıt (2.1.1.2) müşterinin talebinin belirtilen periyot sınırından ($w$) önce bir araca atanmasını ve gönderilmesini önlemek içindir. Kısıt (2.1.1.3) T periyodunda aracın taşıdığı yük miktarı ile ilgilidir. Kısıt (2.1.1.4) j müşterisinin ilgili talebinin parsiyel taşınan miktarı ile ilgilidir. Kısıt (2.1.1.5) aracın yükünün aracın kapasitesini aşmasını

önlemektedir. Kısıt (2.1.1.6) farklı bölgelerdeki müşterilerin taleplerinin aynı periyotta aynı araç ile gitmesini engellemektedir. Kısıt (2.1.1.7) bir aracın bir periyotta birden fazla bölgeye gidemeyeceğini belirtmektedir. Kısıt (2.1.1.8)-( 2.1.1.10) arası işaret kısıtlarıdır.

### 2.1.2. Uzaklık Bazlı Model

$$Min\ z = \sum_{v\in V}\sum_{t\in T} dc_v dist_{vt} + \sum_{j\in M}\sum_{i\in K}\sum_{t\in T} R_{jit}p_{ij} + \sum_{i\in K}\sum_{V\in V}\sum_{t'=1}^{T}\sum_{t=t'}^{T}\sum_{j\in M}\sum_{i\in K} X_{jit'vt}S_v$$

$$\sum_{v\in V}\sum_{t'=1}^{t} X_{jit'vt} \le 1 \qquad \forall j \in M;\ \forall i \in K;\ \forall t \in T \qquad (2.1.2.1)$$

$$\sum_{v\in V}\sum_{t'=1}^{t-w} X_{jit'vt} = 0 \qquad \forall j \in M;\ \forall i \in K;\ \forall t \in T \qquad (2.1.2.2)$$

$$L_{vt'} = \sum_{j\in M}\sum_{t=t'}^{T}\sum_{i\in K} X_{jit'vt}d_{jit} \qquad \forall t' \in T;\ \forall v \in V \qquad (2.1.2.3)$$

$$R_{jit} = d_{jit} - \sum_{v\in V}\sum_{t'=1}^{t} X_{jit'vt}d_{jit} \qquad \forall j \in M;\ \forall i \in K;\ \forall t \in T \qquad (2.1.2.4)$$

$$L_{vt} \le \sum_{b\in B} y_{vbt}Cap_v \qquad \forall v \in V;\ \forall t \in T \qquad (2.1.2.5)$$

$$\sum_{t=t'}^{T}\sum_{i\in K}\sum_{j\in B_b} X_{jit'vt} \le y_{vbt'}M \qquad \forall b \in B;\forall v \in V;\ \forall t' \in T \qquad (2.1.2.6)$$

$$dist_{vt} \ge D_j X_{jitvt'} \qquad \forall v \in V;\ \forall j \in M;\ \forall i \in K;\ \forall t \in T;\ \forall t' \in T : t' \ge t \qquad (2.1.2.7)$$

$$\sum_{b\in B} y_{vbt} \le 1 \qquad \forall v \in V;\ \forall t \in T \qquad (2.1.2.8)$$

$X_{jit'vt} \in \{0,1\}$ $\forall j \in M;\ \forall i \in K;\ \forall t \in T;\forall v \in V;\ \forall t \in T$ (2.1.2.9)

$y_{vbt} \in \{0,1\}$ $\forall v \in V;\ \forall b \in B;\forall t \in T$ (2.1.2.10)

$L_{vt} \geq 0$ $\forall v \in V;\forall t \in T$ (2.1.2.11)

$R_{vit} \geq 0$ $\forall v \in V;\forall i \in K;\ \forall t \in T$ (2.1.2.12)

Yukarıdaki modelin bir önceki model ile iki farkı bulunmaktadır. İlk olarak uzaklık bazlı modelde amaç fonksiyonu hesaplanırken aracın bölgeye gitme maliyeti yerine aracın aynı periyotta gittiği müşteriler arasından maksimum mesafe hesaplanıp maliyete yansıtılmaktadır. İkinci fark ise uzaklık bazlı modelde bölge bazlı modele ek olarak Kısıt (2.1.2.7) eklenmiştir. Kısıt (2.1.2.7) v aracının t periyodunda gönderildiği maksimum mesafenin hesaplanması ile ilgilidir.

# 3. LİTERATÜR ARAŞTIRMASI

Literatür araştırması bölümünde tez kapsamında çalışılan problem ile ilgili olduğu düşünülen üç problem türünden ve sabitle&optimize et sezgisel metodundan bahsedilecektir. Araştırıldığı üzere, tez kapsamında çalışılan problem yapısına literatürde henüz rastlanmamıştır. Fakat ilgili olduğu düşünülen problemler üretim planlama, envanter rotalama ve kutulama problemleridir.

Bu çalışmada çözülmeye çalışılan problemin tam karşılığı literatürde bulunamamıştır. Çalışılan problemin bölgeleri tek bölge, araçları tek tip araç, sıfır ziyaret başına maliyet ve parsiyel taşımacılık maliyeti yeterince yüksek olarak alındığında problem literatürdeki kutulama problemine dönüşmektedir. Kutulama problemlerinin NP-tam problem sınıfından olması sebebiyle tez kapsamında çalışılan problemin NP-zor sınıfına ait bir problem olduğu söylenebilmektedir.

İlgili olan bir diğer problemse envanter rotalama problemidir. Envanter rotalama problemi tez kapsamında çalışılan problemin tek periyotluk ve maliyetin sadece rotalama maliyetlerinden oluşan versiyonudur.

Tez kapsamında çalışılan problemin sisteminde de tek üretici ve çok müşteri bulunduğu göz önünde bulundurularak benzerliklerin ve maliyet yapılarının araştırılması açısından taşıma problemlerinden bazı makaleler incelenmiştir. Samet dov vd. [21] 1984 yılında taşıma problemleri için olası maliyetleri incelemişlerdir. Taşıma problemlerinin başlıca maliyetleri aşağıdaki gibi sıralanmaktadır.

**Sabit Maliyetler**

- Rota başına sabit taşıma maliyeti
- Rota kullanmanın sabit maliyeti
- Ürün taşırken kullanılan aracın sabit maliyeti
- Tüketim merkezinden üretim merkezine aracı göndermenin sabit maliyeti
- Ürünü göndermenin rotaya göre sabit maliyeti
- Durma başına maliyet

- Kullanılan taşıma sisteminin sabit maliyeti
- Orjini bir kez kullanmanın maliyeti

**Değişken Maliyetler**

- Ürünü göndermenin birim maliyeti
- Uzaklığa bağlı birim maliyet
- Ağırlığa bağlı birim maliyet
- Bir birim ürünün dağıtım zamanına bağlı maliyet
- Bir birim ürünün hacmine bağlı maliyet
- Aracı kullanacak sürücüye bağlı maliyet
- Bir birim ürünün üretim ve taşıma maliyeti
- Aracın km başına maliyeti

Bir diğer ilgili konu sabitle ve optimize et metodudur. Bu çalışmada önerilen çözüm yaklaşımı bu metodun uyarlamasıdır. Dolayısıyla aşağıda bu üç problem üzerine literatürden bazı çalışmalar açıklanmıştır.

### 3.1. Kutulama Problemleri

$(v \in V)$ hacmine sahip $B$ adet $(s \in S)$ kutularına, $n$ adet $a_1..a_n$ boyutlarına sahip nesnelerin yerleştirilmesiyle alakalı problemlemlerdir. Nesneler yerleştirilirken amaç en az sayıda kutuyu kullanmaktır. Kutulama problemleri tek boyutlu, iki boyutlu, lineer yükleme, ağırlığa göre yükleme, maliyete göre yükleme gibi kendi içinde çeşitlere ayrılmaktadır ve birçok çeşidi bulunmaktadır[24]. Kutulama problemleri genellikle NP-Zor problemlerdir[6,22]. Aşağıda kutulama problemleri için incelenen makalelerden bahsedilmiştir.

Coffman vd. [6] NP-Zor problemler için yaklaşımlar önermişlerdir. Kutulama problemleri için iyi çözüm veren bir algoritmik yaklaşım getirmişlerdir. Bu makalede kutulama problemlerinin yapısı incelenmiş ve tez kapsamında çalışılacak problemin kutulama probleminin özel bir durumu olduğu belirlenmiştir. Kutulama

problemlerinin NP-Zor problem olduğu makale sayesinde öğrenilmiş ve çalışılacak problemin kompleksitesinin de NP-Zor olduğu anlaşılmıştır.

Karmarkar vd. [14] 1982 yılında tek boyutlu kutulama problemleri için etkili bir yaklaşım geliştirmişlerdir. Geliştirdikleri yöntem ile kutulama problemleri için optimale yakın sonuçlar elde etmişlerdir ve algoritmalarının performansını örnek problemler ile kanıtlamışlardır. Fleszar vd. [8] tek boyutlu kutulama problemleri için 1370 problemin 1329'unda optimal çözümü elde eden hibrid bir algoritma geliştirmişlerdir. 2002 yılında yayınladıkları makale ile algoritmalarının 1982'de Karmarkar vd. [14] tarafından yapılan algoritmadan daha iyi sonuçlar verdiğini kanıtlamışlardır. 1370 problemin 1329'unda optimal çözümle aynı değeri veren algoritmaları kutulama problemleri için en iyi alternatiflerden biri olarak bilinmektedir. Scholl vd. [22] tek boyutlu kutulama problemleri için BISON isimli çözüm algoritması geliştirmişlerdir. BISON algoritması tabu arama algoritması ve dal-sınır algoritmasını birlikte çalıştırmaktadır. Sayısal çıktıların oldukça iyi olduğu görülmüştür. Kröger [15] bidon yükleme problemleri için aynı problem çözümü için genetik algoritmayı incelemiştir.

### 3.2. Envanter Rotalama Problemleri

Dünya üzerindeki firmaların tedarik zinciri performansını önemsemeye başlamasıyla ön plana çıkmış problemlerdir. Envanter rotalama problemleri genellikle tek bir ürünün, tek bir merkezden $n$ adet müşteriye $T$ zaman uzunluğuna sahip planlama ufkunda en az maliyetle ulaştırılmasını incelemektedir. $i$ müşterisinin planlama ufku içerisinde belirli aralıklarla $v_i$ oranında ürün tüketimi ya da talebi bulunmaktadır. Fakat stokladığı miktar talebi maksimum $C_i$ miktarına kadar çıkabilmektedir. Başlangıçta müşteride üründen belirli derecede $I_i$ miktarında stok bulunabilir. Her biri $Q$ kapasiteli $m$ adet araca sahip üretici ürününü müşterilere belirlenen bu planlama çevreni içerisinde en az dağıtım maliyeti ile ulaştırmaya çalışmaktadır[4,5,23].

Bu dağıtım yapılırken genellikle 3 karar verilmektedir;

- Hangi dağıtım rotalarının kullanılacağı
- Müşteriye o anda hangi miktarda ürün gönderileceği
- Müşteriye dağıtımın ne zaman yapılacağı

Envanter rotalama problemleri genellikle araç rotalama problemlerine benzetilmektedir ancak envanter rotalama problemleri müşterinin kullanım oranıyla ilgilenir. Araç rotalama problemlerinde ise müşterinin istekleri ön plandadır. Envanter rotalama problemleri (IRP) genellikle deterministik ve statik bir yapıya sahiptir çünkü genelde müşterinin kullanım oranı sabit ve bilinmektedir. Ancak gerçek hayatta müşterilerin talepleri ve kullanım oranları tahmin edilemeyeceğinden gerçek hayattaki sistem stokastik ve dinamik bir yapıya sahiptir. Gerçek hayat problemine yakınlıktaki arayış ortaya stokastik envanter rotalama problemlerinin ortaya çıkmasını sağlamıştır (SIRP). Stokastik envanter rotalama problemlerinde müşteri talepleri bilinmezken envanter rotalama problemlerinde müşteri taleplerinin bilindiği varsayılır. Bu çalışmadaki probleme araç rotalama yapılmamaktadır ve ürünlerin gideceği önceden belirlenmiş rotalar yoktur. Problem tanımında bahsedildiği gibi müşterilerin coğrafik olarak yerleştiği bölgeler bilinmektedir ve bu bölgelere araç göndermenin belirli maliyeti vardır. Rota kullanılmaktadır[5,12,17,23].

Moin vd. [17] envanter rotalama problemlerine lojistik olarak genel anlamda incelemişlerdir ve envanter rotalama problemleri için geniş bilgiler sunmuşlardır. Coelho vd. [4,5] envanter rotalama problem türlerini incelemişlerdir ve bu problem türleri için çözüm yöntemleri türetmişlerdir. Hvattum vd. [12] stokastik envanter rotalama problemlerini incelemişlerdir ve bu problemler için senaryo ağaçları oluşturmuşlardır.

### 3.3. Sabitle ve Optimize Et Metodu

Tez kapsamında çalışılacak problemin çözümü için sabitle & optimize et metodunun performansı değerlendirileceğinden daha önceden başka problemerde uygulanan sabitle & optimize et problemleri için literatür araştırmasına gidilmiştir. Sabitle & optimize et metodunu 2005 yılında Gintner vd. [9] tarafından otobüs çizelgeleme probleminde uygulanmıştır. Bu makale ile sabitle & optimize et metodunun uygulanış biçimi ve aşamaları öğrenilmiş ve tez kapsamında çalışılacak probleme nasıl uygulanacağı belirlenmiştir. Pochet vd. [19] 2006 yılında yayınladıkları makale ile sabitle & optimize et metodunu üretim planlama problemleri üzerinde uygulamışlardır. Federgruen vd. [7] 2007 yılında yayınladıkları makale ile sabitle & optimize et metodunun uygulanışını çok ürünlü, kapasiteli parti büyüklüğü belirleme problemleri üzerinde göstermişlerdir. Helber vd. [11] 2010 yılında yayınladıkları makale ile aynı algoritmanın çok seviyeli, kapasiteli parti büyüklüğü belirleme problemi için uygulanışını anlatmışlardır.

Sabitle & optimize et metodunun tanımı ve aşamaları yukarıda araştırılan makaleler ile öğrenilmiştir ve tez kapsamında çalışılan probleme uygulanması uygun görülmüştür. Uygulanan çözüm metodu başlığı altında anlatılacaktır.

# 4. UYGULANAN ÇÖZÜM YÖNTEMİ

Tek üreticili, çok müşterili sistemde taşıma planlaması probleminin matematiksel modeli ILOG CPLEX çözücüsü ve Eclipse Java ile kodlanmıştır. Ancak problemin NP-Zor olması sebebi ile problemin sezgisel olarak çözülmesine karar verilmiş ve iki aşamadan oluşan sabitle ve optimize et metodu problem üzerinde 4 farklı şekilde uygulanmıştır. İlk aşamada sabitle ve optimize et metodu problemi belirlenen bir parametreye göre belli sayıda karar verme problemlerine bölmektedir ve bu problemleri sırası ile çözerek sonunda ana problem için bir geçerli çözüm elde etmektedir. İkinci aşamada ise birinci aşamada bulunan çözümden yararlanılarak amaç fonksiyonunun iyileştirilmesine çalışılır. Amaç fonksiyonu iyileştirilmeyene kadar veya belirlenen adım sayısı kadar ikinci aşamaya devam edilir. Geliştirilen yöntemin ilk aşaması altı adımdan oluşmaktadır. İlk adımda belirlenen parametreye göre model $Z$ adet lineer problem grubuna bölünür. Daha sonra ikinci adımda bölünen lineer problemlerden ilkinin değişkenleri tamsayılı bırakılır. Geriye kalan $(Z-1)$ adet problemin değişkenleri ise gevşetilmiş olarak bırakılır ve ana problem modeli bu şekilde çözülür. Üçüncü adımda ilk gruptaki değişkenlerin tamsayılı çözüm değerleri sabitlenir, ikinci gruptaki değişkenler tamsayılı yapılır. Kalan $(Z-2)$ adet problemin değişkenleri gevşetilmiş olarak bırakılır ve ana problem modeli bu şekilde çözülür. Bu şekilde Z grubun tamamı için tamsayılı çözüm elde edilene kadar devam edilir. Önerilen yöntemin ilk aşamasının adımları aşağıda gösterilmiştir;

## 4.1. İlk Aşama Adımları

***Adım 1:*** *Hangi parametreye göre ikili karar değişkenlerinin gruplara ayrılacağını belirle. Karar değişkenlerini bir parametreye göre Z adet gruba ayır.* $S_i$ *, i. gruptaki ikili karar değişkenlerinin kümesi olsun.*

***Adım 2:*** *İterasyon sayacı i'yi bire set et (i=1).*

***Adım 3:*** *Eğer i>1 ise, k=1..i-1 için $S_k$'daki ikili değişkenleri (i-1). iterasyondaki değerlerine sabitle.*

***Adım 4:*** *k=i+1..Z için $S_k$'daki ikili değişkenleri 0-1 aralığına gevşet.*

***Adım 5:*** *$S_i$'deki değişkenleri ikili varsay.*

***Adım 6:*** *Modeli çöz ve i değerini i=i+1 olarak güncelle. $i \leq Z$ ise adım 3'e geri dön. i>Z ise ilk aşama tamamlanmıştır.*

Geliştirilen yöntemin ilk aşaması geçerli bir mümkün çözüm bulur ve aşama ikide iyileştirilmeye çalışılacak amaç fonksiyon değerini verir. Geliştirilen yöntemin ilk aşamasının adımları bir akış diyagramında şekil 4.1'deki gibi gösterilmiştir.

Geliştirilen yöntemin ikinci aşaması iki adımdan oluşmaktadır. Bu aşamada ilk aşamanın sonunda bulduğumuz ana problem için geçerli olan mümkün çözümün amaç fonksiyonunun iyileştirilmesi amaçlanır. İkinci aşamada Z adet problem grubundan hiçbiri gevşetilmiş olarak bırakılmaz. İkinci aşamanın ilk adımında $i = 1$ yapılır ve ilk adıma geçilir. Daha sonra i. grup için problemin ilk aşamanın son adımında bulunan çözümü unutturulur ve değişkenleri tamsayılı yapılır. Bu işlemden sonra i. grup hariç kalan (Z-1) adet grup problemin değişken değerleri ilk aşamanın son adımında bulunan mümkün çözüm değerlerine sabitlenir. İlk adımda ana problem bu şekilde çözdürülür ve ikinci adıma gidilir.

İkinci adımda i'nin yeni değeri i+1 olarak güncellenir. i. grubun değişkenleri tamsayılı yapılırken kalan (Z-i) adet problem grubunun değişken değerleri ikinci adımın ilk aşamasında bulunan çözüme sabitlenir. $(Z - i) = 0$ ise problem bu şekilde çözdürülüp amaç fonksiyonunda iyileşme var mı diye bakılır. Amaç fonksiyonunda iyileşme var ise ilk adıma geri dönülür ve ikinci aşama aynı şekilde amaç fonksiyonu değişmeyene kadar yapılmaya devam edilir.$(Z - i) > 0$ ise ana problem bu şekilde çözülür ve adım ikiye geri dönülür. Belirtilen adımlar aşağıda ve şekil 4.2'deki gibi gösterilmiştir.

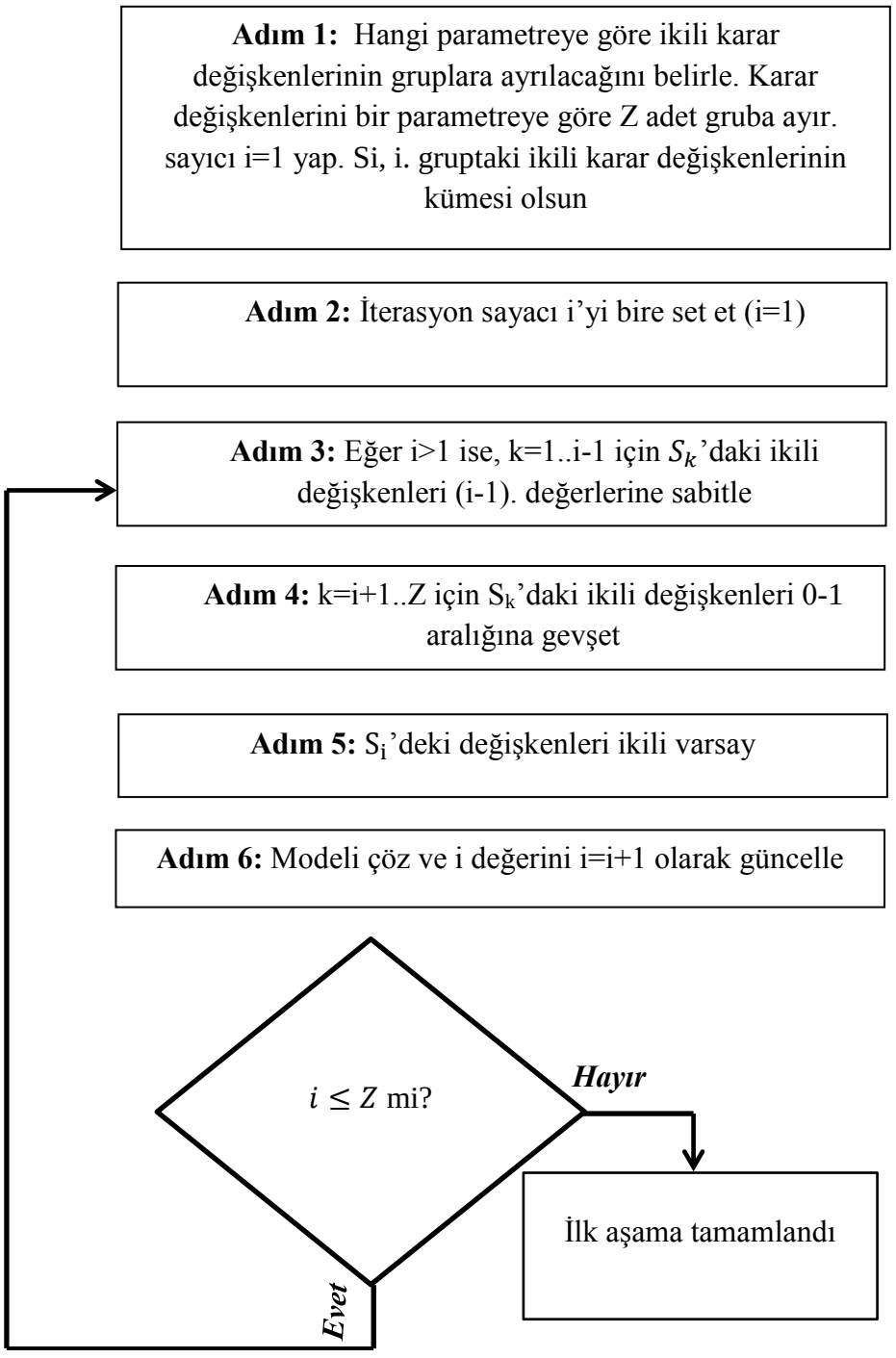
### 4.2. İkinci Aşama Adımları

***Adım 1:*** *i değerini* $i = 1$ *olarak güncelle.*

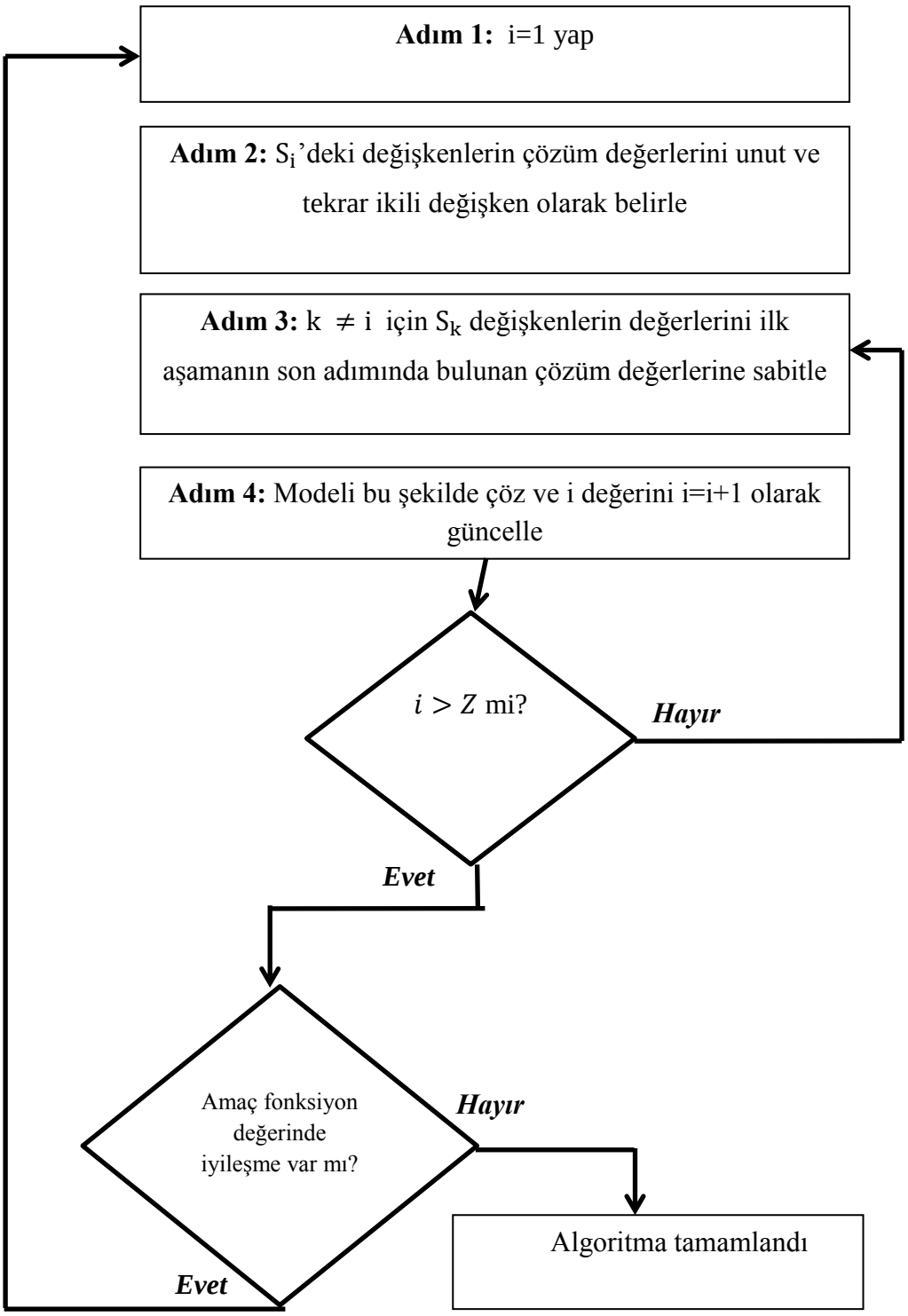
***Adım 2:*** $S_i$*'deki değişkenlerin çözüm değerlerini unut ve tekrar ikili değişken olarak belirle.*

***Adım 3:*** $k = 1,2..Z$ *ve* $k \neq i$ *için* $S_k$ *değişkenlerin değerlerini bulunan çözüm değerlerine sabitle.*

***Adım 4:*** *Modeli bu şekilde çöz ve i değerini i=i+1 olarak güncelle.* $i \leq Z$ *ise adım 2'ye dön. i>Z ise bulunan son çözümü i=1 ikenki çözüm ile karşılaştır. Amaç fonksiyonunda iyileşme var ise adım 0'a geri dön. İyileşme yok ise amaç fonksiyonu daha fazla iyileştirilemez, algoritma tamamlanmıştır.*




Şekil 4.1: Geliştirilen çözüm yönteminin ilk aşamasının şematik gösterimi




Şekil 4.2: Geliştirilen çözüm yönteminin ikinci aşamasının şematik gösterimi

### 4.3. Sabitle ve Optimize Et Metodunun Uygulanış Örneği

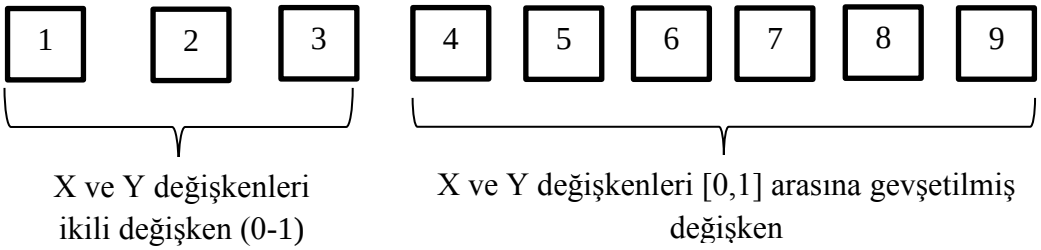
Bu tez kapsamında sabitle ve optimize et metodu farklı parametreler baz alınarak matematiksel modele uygulanmıştır. Uygulanan farklı yöntemler aşağıdaki gibi sıralanmaktadır;

- Çoklu periyot bazlı yaklaşım
- Tek periyot bazlı yaklaşım
- İki periyotlu kayan pencere bazlı yaklaşım
- Araç – periyot bazlı yaklaşım
- Büyük araç öncelikli ve periyot bazlı yaklaşım

#### 4.3.1. Çoklu Periyot Bazlı Yaklaşım

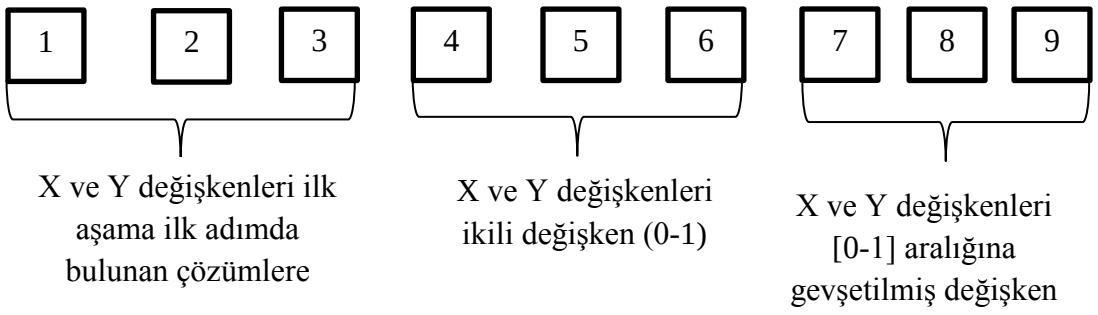
Modeldeki değişkenlerin periyot indislerine göre gruplara ayrıldığı uygulama biçimidir. Problem belirlenen aralıkta değişkenlerin periyot indisine göre bölünmesiyle küçük problemlere ayrılır ve bahsedilen algoritmanın iki aşamalı adımları uygulanır. Örnek olarak 9 periyotluk bir modelin aralık boyutu 3 seçilerek sabitle ve optimize et metodunun uygulanışı anlatılacaktır.

Öncelikle problem bölme boyutu 3 seçildiğinden ana problem 3'er periyotluk 3 problem grubuna bölünecektir. Birinci grup problemde 1. ila 3. periyotlar, ikinci grup problemde 4. ila 6. periyotlar, üçüncü grup problemde 7. ila 9. periyotlardaki değişkenler içerilecektir. Problem bölündükten sonra sabitle ve optimize et metodunun ilk aşaması artık uygulanabilmektedir. Sabitle ve optimize et metodunun ilk aşamasının ilk adımının uygulanışı şekil 4.3'te gösterilmiştir.



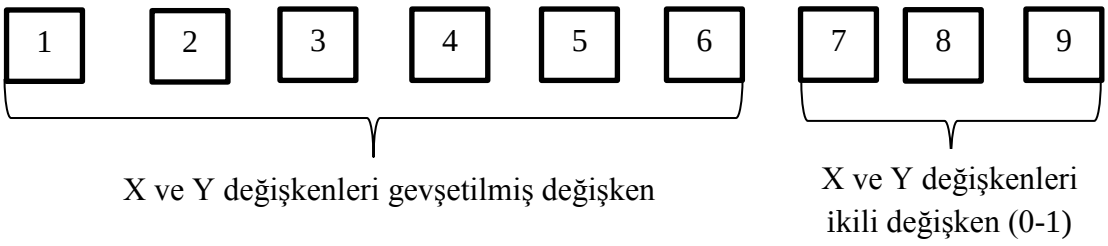

Şekil 4.3: Sabitle ve Optimize et metodunun ilk aşamasının ilk adımının şematik gösterimi

Şekilde görüldüğü gibi, ilk grubun karar değişkenleri ikili değişken kalırken modelin 4. ila 9. periyotlardaki değişkenleri gevşetilmiş olarak bırakılır. Ana problem bu şekilde çözüldükten sonra ilk grup için tamsayılı çözüm bulunur ve ilk aşamanın ikinci iterasyonuna geçilir. İkinci aşamada 1. ila 3. periyottaki çözümler ilk aşamanın ilk iterasyonunda bulunan tamsayılı çözümlere sabitlenir. Algoritmanın ilk sabitleme işlemi bu adımda yapılır. Daha sonra 4. ila 6. periyotları içeren ikinci grubun değişkenleri tamsayılı bırakılır, 6. ila 9. periyotları içeren üçüncü grubun değişkenleri gevşetilmiş olarak bırakılır. İlk aşamanın ikinci iterasyonu şekil 4.4'te şematik olarak gösterilmiştir.




Şekil 4.4: Sabitle ve Optimize et metodunun ilk aşamasının ikinci adımının şematik gösterimi

İlk aşamanın ikinci adımında ana problem değişkenleri yukarıda görülen şemadaki hali alır. Bu şekilde ikinci lineer grubun çözümü elde edildikten sonra 7. ila 9. periyotları içeren grubun çözümü için ilk aşamanın üçüncü adımına gidilir. Üçüncü adımda ilk aşamanın ikinci iterasyonunda bulunan çözümden yararlanılarak 1. ila 6. periyotları içeren iki grubun çözümü yine sabit bırakılmaktadır. Son grup olan ve 7. ila 9. periyotlar arasını içeren grubun değişkenleri ise tamsayılı bırakılacaktır. Üçüncü iterasyon ilk aşamanın son adımı olarak tanımlanmaktadır. Bunun nedeni, $Z = 3$ olmasıdır. Üçüncü iterasyondan sonra artık değişkenlerin tamsayılı yapılıp çözüleceği bir grup bulunmayacaktır ve 3 grup problemin de tamsayılı çözümleri bulunmuş olacaktır. İlk aşamanın son adımında ana problem için mümkün bir çözüm bulunur. Bunun nedeni tüm değişkenlerin artık tamsayılı değerlere sahip olmasıdır. İlk aşamanın son iterasyonunun şematik gösterimi şekil 4.5'te gösterilmiştir.

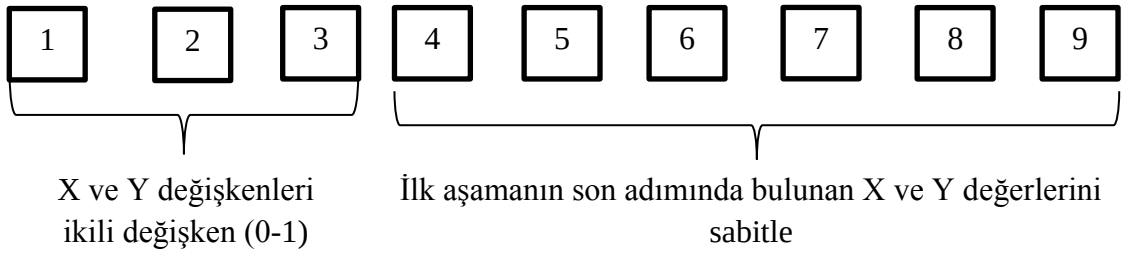



Şekil 4.4: Sabitle ve Optimize et metodunun ilk aşamasının üçüncü adımının şematik gösterimi

Sabitle ve optimize et metodunun ilk aşamasında bulunan çözümün geliştirilmesi için algoritmanın ikinci aşamasına gidilir. İkinci aşamada bulunan her çözüm tamsayılı değişken değerlerine sahip olacaktır ve ikinci aşamaya amaç fonksiyon değeri değişmeyene kadar devam edilecektir.

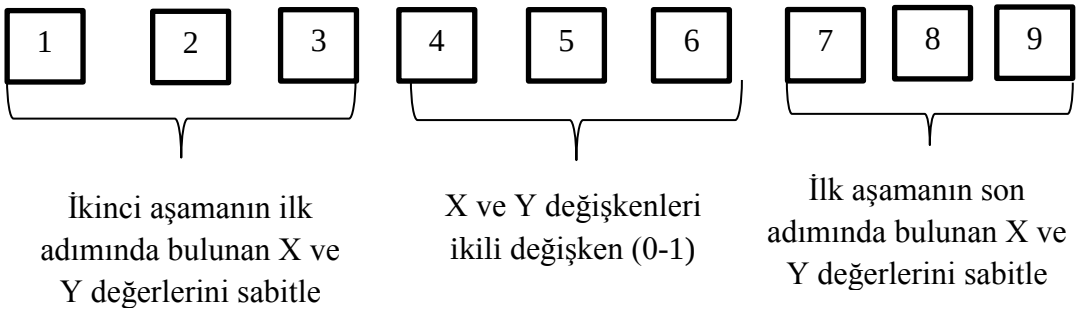
Algoritmanın ikinci aşamasının ilk iterasyonunda ilk grubun sabitlenen değerleri unutulur ve yeniden çözülmesi amacıyla bu grubun değişkenleri tamsayılı yapılır.

Kalan iki grubun değişkenleri algoritmanın ilk aşamasında bulunan mümkün çözüme sabitlenir. Algoritmanın ikinci aşamasının ilk iterasyonu şekil 4.6'da gösterilmiştir.




Şekil 4.6: Sabitle ve Optimize et metodunun ikinci aşamasının ilk adımının şematik gösterimi

Model şekil 4.6'da görülen şekilde çözüldükten sonra algoritmanın ikinci aşamasının ikinci iterasyonuna geçilir. Bu iterasyonda ilk ve üçüncü grubun çözümleri bir önceki iterasyonda bulunan çözümlere sabitlenir. İkinci grubun çözüm değeri unutularak değişkenleri tamsayılı varsayılır ve model çözülür. Algoritmanın ikinci aşamasının ikinci iterasyonu şekil 4.7'de gösterilmiştir.




Şekil 4.7: Sabitle ve Optimize et metodunun ikinci aşamasının ikinci adımının şematik gösterimi

Model şekil 4.7'de görülen şekilde çözüldükten sonra algoritmanın ikinci aşamasının üçüncü iterasyonuna geçilir. Bu adımda ilk grubun değerleri ikinci aşamanın ilk iterasyonunda bulunan değerlerine sabitlenir, ikinci grubun değerleri ise ikinci aşamanın ikinci iterasyonunda bulunan değerlere sabitlenir. Üçüncü grubun değişkenleri tamsayılı varsayılır ve model bu şekilde çözülür. Üçüncü iterasyondan sonra tamsayılı çözülecek başka grup kalmaması nedeniyle, amaç fonksiyonundaki iyileşmeye bakılır. Üçüncü iterasyon sonunda amaç fonksiyonunda iyileşme var ise, algoritmaya ikinci aşamanın ilk iterasyonundan yeniden başlanarak amaç fonksiyon değeri değişmeyene kadar aynı adımlarla devam edilir. Üçüncü iterasyon sonunda amaç fonksiyonu değeri ilk aşamanın son iterasyonunda bulunan amaç fonksiyon değerine göre iyileşmedi ise bu durum amaç fonksiyon değerinin daha fazla iyileştirilemeyeceğini göstermektedir. Bu durumda algoritma bitirilir ve geçerli çözüm en iyi çözüm olarak bulunmuş olur. Bölünerek parça parça çözülen problemlerde ana problem için geçerliliğin sağlanma sebebi parsiyel taşıtma opsiyonudur. Bu nedenle tüm kısıtları sağladığı zaten bilinmektedir. Algoritmanın 4 farklı uygulanış yönteminden biri olan çoklu periyot bazlı uygulanış bu bölümde örnek şemalar ile anlatılmıştır.

### 4.3.2. Tek Periyot Bazlı Yaklaşım

Bu uygulanış yönteminin amacı çok büyük problemlerin çözümünü kolaylaştırmak içindir. Tek periyot bazlı uygulanışın çoklu periyot bazlı uygulanıştan tek farkı ana problemin periyot indisine göre tek periyotluk gruplara bölünmesidir. Böylece problem parametresinde periyot sayısı kaç olarak belirlendiyse o kadar grup problem oluşacaktır.

Örnek olarak 9 periyotluk problem düşünüldüğünde, problem kendi içinde 9 gruba bölünecektir. Dolayısı ile ilk aşamanın tamamlanması 9 iterasyon sürecektir. İlk aşamanın ilk adımında birinci periyot için değişkenler tamsayılı yapılır, kalan 2. ila 9. periyotlar arası değişkenler gevşetilmiş olarak bırakılır. Model bu şekilde

çözüldükten sonra ilk aşamanın ikinci iterasyonuna gidilir. İkinci iterasyonda ilk iterasyondan bulunan 1. periyottaki tamsayılı değerler sabitlenir. 2. periyot için değişkenler tamsayılı bırakılır ve kalan 3. ila 9. periyotlar arası değişkenler gevşetilmiş olarak bırakılır. Benzer adımlar izlenerek ilk aşamaya $i = Z$ denklemi sağlanana kadar devam edilir. İlk aşama sonunda bulunan ve ana problem için geçerli olan mümkün çözüm algoritmanın ikinci aşamasında iyileştirilmeye çalışılır.

İkinci aşamanın ilk iterasyonunda ilk periyot için değişken değerleri unutulur ve tamsayılı değişken olarak bırakılır. Kalan 8 periyodun değerleri ilk aşamanın son iterasyonunda bulunan değerlere sabitlenir ve ana problem tekrar çözdürülür. Çözdürüldükten sonra ikinci aşamanın ikinci iterasyonuna geçilir. İkinci iterasyonda ilk periyodun değişken değerleri ikinci aşamanın ilk iterasyonunda bulunan tamsayılı değerlere sabitlenir. İkinci periyodun değişken değerleri unutulur ve tamsayılı bırakılır. Kalan 3. ila 9. periyotların değerleri yine algoritmanın ilk aşamasının son iterasyonunda bulunan çözüme sabitlenir. Ana problem bu şekilde çözdürüldükten sonra algoritmanın üçüncü aşamasına geçilir.

Benzer adımlarla algoritmanın ikinci aşamasının 9. iterasyonuna gelinir. 9. iterasyon sonunda amaç fonksiyon değerinde ilk aşamanın son iterasyonundaki amaç fonksiyonuna göre iyileşme bulundu ise algoritmanın ikinci aşamasına en baştan başlanır ve amaç fonksiyonunda iyileşme olmayana kadar devam edilir. 9 iterasyon sonunda amaç fonksiyon değerinde iyileşme bulunamadı ise, bu durum amaç fonksiyon değerinin daha fazla iyileştirilemeyeceğinin sinyalidir ve algoritmanın ikinci aşaması tamamlanmış sayılmaktadır.

Algoritmanın ikinci uygulanış yöntemi olan tek periyot bazlı uygulanış anlatılan şekilde uygulanır ve daha büyük problemleri çözmek için kullanılır. Parametre değerlerinin gerçek hayata yakın (çok büyük) olduğu problemlerde, problem periyot baz alınarak küçük problemlere bölünse bile bölünerek elde edilen problemlerin boyutları hala çok büyük olabilmektedir. Bu durumda çoklu periyot bazlı uygulanış çözüm süresini uzatabilir veya yeteri kadar sürede çözüm vermeyebilmektedir. Tek periyot bazlı uygulanışta problem tek periyotlar halinde çözüleceği için bölünerek elde edilen problemlerin boyutları daha küçük olmaktadır. Bu durum daha büyük

problemlerde tek periyot bazlı uygulanış yönteminin çalışmasını kolaylaştırmaktadır. Ancak problemi tek periyotlar halinde çözmenin çözüm kalitesi itibariyle çoklu periyot yaklaşımına göre dezavantajlı olacağı bellidir.

### 4.3.3. İki Periyotlu Kayan Pencere Yaklaşımı

Bu yöntem çoklu periyot bazlı yönteme alternatif olarak sunulmuştur. Problem ikili periyotlar halinde ancak her adımda bir periyot sonrasından başlayarak çözülmektedir. Yöntemin ilk aşamasının ilk adımında 1. ve 2. periyot değişkenleri tamsayılı, kalan periyotlardaki değişkenler gevşetilmiş değişken olarak bırakılır. Problem bu şekilde çözüldükten sonra ikili periyot penceresi bir periyot sağa kaydırılır. Böylece ilk aşamanın ikinci adımında 2. ve 3. periyotlar arası değişkenler tamsayılı, kalan periyottaki değişkenler gevşetilmiş olarak bırakılır. Belirlenen penceren önceki periyotlar önceki adımda bulunan çözümüne sabitlenir. İlk aşamanın ikinci adımında belirlenen periyot penceresinden önce sadece 1. periyot bulunmaktadır. 1. Periyottaki değişkenler ilk adımda bulunan çözüm değerlerine sabitlenir. Model bu şekilde çözülür ve üçüncü adıma geçilir. Benzer şekilde periyot penceresi son periyodu kapsayana kadar kaydırılmaya devam edilir. İlk aşamanın son adımında diğer yöntemlerdeki gibi gevşetilmiş bir değişken kalmayacağı için ana problem için geçerli bir çözüm bulunur.

İkinci aşamada benzer adımlarla bulunan geçerli çözüm değeri iyileştirilmeye çalışılır. İki periyotlu kayan pencere yönteminin problemi tek periyotluk gruplar halinde çözmemesi sebebiyle iyi bir performans vermesi beklenmektedir. İki periyotlu kayan pencere yönteminde çözülen problem boyutu tek periyot bazlı yaklaşıma göre daha büyük, çoklu periyot bazlı yaklaşıma göre daha küçüktür. Ancak çok periyotluya nazaran her adımda sadece bir periyodun değişkenleri sabitlenmektedr. Bu nedenle iki periyotlu kayan pencere yönteminin tek periyot bazlı yaklaşımdan daha uzun çoklu periyot bazlı yaklaşımdan daha kısa sürede çözüm vermesi beklenmektedir.

### 4.3.4. Araç – Periyot Bazlı Yaklaşım

Araç – periyot bazlı yaklaşım, ana problemin sadece belirlenen periyot aralığına göre değil hem periyot hem de araç parametresi baz alınarak küçük problemlerin elde edildiği uygulanış biçimidir. Örnek olarak diğer iki yöntemdeki gibi 9 periyotluk ve 3 büyük 3 küçük olmak üzere toplam 6 aracın bulunduğu bir problem düşünülürse, yöntemin ilk aşamasının ilk iterasyonunda 1. periyottaki küçük araçların değişkenleri tamsayılı yapılır ve kalan tüm araçların değişkenleri gevşetilmiş olarak bırakılır. Ana problem belirtilen şekilde çözdürüldükten sonra ilk aşamanın ikinci iterasyonuna gidilir. İkinci iterasyonda 1. periyotta küçük araçlar için bulunan tamsayılı değişken değerleri sabitlenir ve 1. periyottaki büyük araçların değişkenleri tamsayılı bırakılır. Kalan 8 periyottaki tüm araçlar için karar değişkenleri gevşetilmiş olarak bırakılır. Ana problem belirtilen şekilde çözüldükten sonra 1. periyotta bulunan tamsayılı çözüm değerleri artık bellidir. İlk aşamanın üçüncü iterasyonunda ilk periyottaki küçük ve büyük araçlar için çözüm sabitlenir ve ikinci periyottaki küçük araçların değerleri tamsayılı yapılır. İkinci periyottaki büyük araçların ve kalan 7 periyottaki tüm araçların değerleri gevşetilmiş olarak bırakılır. Ana problem belirtilen şekilde çözülür ve ilk aşamanın dördüncü iterasyonuna geçilir. Dördüncü iterasyonda ilk periyottaki değerler hala sabitken, bu sabitlemeye ek olarak ikinci periyottaki küçük araçların hangi bölgeye gideceği eklenir. İkinci periyottaki küçük araçların hangi bölgeye gideceğinin bilgisi üçüncü iterasyon sonucunda bulunmuştur. Dördüncü iterasyonda ikinci periyottaki büyük araçların değişkenleri tamsayılı yapılır ve kalan 7 periyottaki tüm araçların değişkenleri gevşetilmiş olarak bırakılır.

Problem belirtilen parametrelerle $Z = 18$ alt probleme bölüneceğinden her adımda değeri 1 artan i sayıcısı 18 olana kadar özetle gevşetilmiş araç değişkeni kalmayana kadar algoritmanın ilk aşamasına devam edilir. Algoritmanın ilk aşamasının son iterasyonunda bulunan mümkün çözüm ikinci aşamada iyileştirilerek ana problemin optimal değerine yaklaştırılmaya çalışılır.

Araç periyot bazlı uygulanışta problemin alt problemlere bölüneceği boyut istenilen taktirde değiştirilebilmektedir ve kolayca adaptasyonu sağlanabilmektedir. Örnek olarak algoritma boyunca araçların tek periyotlar yerine ikili, üçlü periyotlar halinde de tamsayılı değerlerinin bulunup sabitlenmesi mümkündür. Ancak bu tez kapsamında araç – periyot bazlı uygulanışın çok büyük boyutlardaki problemlerde performansının değerlendirilmesi amaçlanmıştır. Bu nedenle ana problem küçük problemlere bölünürken tek periyotluk araç sabitleme yaklaşımı kullanılmıştır.

### 4.3.5. Büyük Araç Öncelikli ve Periyot Bazlı Yaklaşım

Ana problem küçük problemlere bölündükten sonra sabitleme işlemleri yapılırken, önce tüm periyotlardaki büyük araçlara öncelik verilir ve daha sonra tüm periyotlardaki küçük araçlar için sabitleme işlemi yapılıp mümkün çözüme ulaşılır. Büyük araçların küçük araçlara oranla daha maliyetli olduğu bilinmektedir. Sabitle ve optimize et algoritması uygulanırken önceliğin büyük araçlara verilmesi, bir önceki kısımda anlatılan araç-periyot bazlı uygulanışa oranla daha iyi amaç fonksiyon değerine sahip mümkün çözümün bulunmasına sebep olmuştur. Bu yöntemin adımları araç – periyot bazlı uygulanışla aynıdır. Bu yöntemde öncelikle her periyot için büyük araçların hangi bölgeye gideceğinin bilgisi bulunur. Daha sonra yine sırayla her periyot için büyük araçların hangi bölgeye gideceğinin bilgisi sabitken küçük araçların hangi bölgelere gideceği bulunur. $i$ sayıcısı $Z$ adet alt problem sayısına eşit olduğunda mümkün çözüm bulunmuştur ve yöntemin ikinci aşamasında bu mümkün çözümden elde edilen amaç fonksiyon değeri iyileştirilmeye çalışılır.

# 5. ÖNERİLEN YÖNTEMLERİN PERFORMANSI

Numerik analiz bölümünde verilerin hangi yöntemlerle ve nasıl üretildiğinden bahsedilecektir. Ayrıca bu bölümde üretilen problem setleri üzerinde yukarıda açıklanan sezgisel çözüm yaklaşımlarının performansı test edilecektir. Yöntemin uygulanacağı problemlerin verilerinin üretilmesinin daha kolay olmasını sağlayan oranlardan ve bu oranların problem zorluğuna olan etkisinden bahsedilecektir.

Çalışılan problemin literatürde tam karşılığına rastlanılmadığı için hazır veri bulunmamaktadır. Bu nedenle veri üretimi çalışma boyunca zorunlu hale gelmiştir. Veriler üretilirken sabitle ve optimize et metodunun optimalle karşılaştırılması amaçlandığı için, veriler gerçek hayat verilerinin boyutlarında değil, optimal sonuca ulaştıracak bilgisayar programı çözücüsünün makul sürede çözebileceği boyutlarda üretilmiştir. Aşağıda bahsedilen oranlar kullanılarak değişik zorluklara sahip 8 setlik ve her setin 10 problem içerdiği 80 problem üretilmiştir. Sezgisel metodun anlatılmış olan 4 farklı uygulanışı 80 adet problemde ayrı ayrı denenmiş böylece 320 adet sonuca ulaşılmıştır.

Numerik analiz bölümünde ulaşılan 320 adet sonuç için sabitle ve optimize et metodu ile bulunan çözüm süreleri ve amaç fonksiyon değerlerinin performansı incelenecektir. Optimal çözümün süreleri ve çözüm değerleri ile karşılaştırılacaktır. Sabitle ve optimize et metodunun 4 farklı uygulanışından çözüm süresi açısından hangisinin en verimli olduğu, amaç fonksiyon değeri açısından hangisinin en iyi olduğu ile ilgili karşılaştırmalar yapılacaktır.

## 5.1. Kullanılan Oranlar

Çalışılan problem için veri üretilirken, veri üretilmesinin kolaylaştırılması için belirli oranlar kullanılmıştır. Bu oranlara verilen değerler sayesinde üretilen problemlerin zorluk dereceleri ve çözülme süreleri değişebilmektedir.

$$\boldsymbol{a} = \frac{Bir\ müşterinin\ bir\ periyotta\ talep\ ettiği\ ortalama\ ürün\ miktarı}{Küçük\ araç\ kapasitesi}$$

$$\boldsymbol{CV} = \frac{Taleplerin\ standart\ sapması}{Ortalama\ talep} = \frac{\sigma}{\mu}$$

$$\frac{Büyük\ araç\ maliyeti}{Büyük\ araç\ kapasitesi} = \boldsymbol{sef}\ \frac{Küçük\ araç\ maliyeti}{Küçük\ araç\ kapasitesi}$$

$a$ oranı küçük araç kapasitesine göre ortalama bir müşterinin talebini belirler. $a$ oranının artması araç başına düşen talebi arttıracağı ve daha fazla araç kullanımına neden olacağı için çalışılan problem zorlaşacaktır. Değişim katsayısı $CV$'nin artması ise taleplerin standart sapmasını arttıracaktır. Standart sapmanın artması taleplerin birbirinden farklılığını ve taleplerin büyüklük-küçüklük oranını arttıracağı için problem yine zorlaşmış olacaktır. Skala ekonomisi faktörü (sef) ise büyük aracın birim kapasite maliyetinin küçük aracın birim kapasite maliyetine göre ne kadar fazla olacağını belirlemektedir.

## 5.2. Veri Üretimi

Ön deneme çalışmaları sonucu 80 problem için parametreler belirlenmiştir. Problem için; 9 müşteri, 3 bölge ve 9 periyotluk planlama ufkunun olması kararlaştırılmıştır. Ürün tipinin tek olması ve 3 büyük 3 küçük araç olmak üzere toplamda lojistik firmalardan kiralanabilecek maksimum araç sayısı 6 olarak belirlenmiştir. İki tip araç

olduğu varsayılmıştır. Büyük araçların kapasitesi 25 ton, küçük araçların kapasitesi 16 ton olarak belirlenmiştir. Küçük aracın en ucuz bölgeye gitmesinin maliyeti 100 TL olarak belirlenmiştir.

İlk set oluşturulurken $a = 0.1$, CV = 0.50, sef = 0.70 alınmıştır. İkinci bölgenin birinci bölgeye %10 daha pahalı olduğu, üçüncü bölgenin ikinci bölgeye göre %10 daha pahalı olduğu varsayılmıştır. Oranlar yukarıdaki gibi alındığında;

$$a = \frac{Bir\ müşterinin\ bir\ periyotta\ talep\ ettiği\ ortalama\ ürün\ miktarı\ (\mu)}{Küçük\ araç\ kapasitesi}$$

$\mu = 0.1\ x\ 16 = 1.6$

Bir müşterinin bir periyotta üründen talep ettiği ortalama miktar 1.6 birim olarak belirlenmiştir.

$$CV = \frac{\sigma}{\mu}$$

$\sigma = \mu\ x\ CV = 1.6\ x\ 0.5 = 0.8$

Taleplerin standart sapması belirlenen CV ve ortalama talebin $(\mu)$ çarpımı ile 0.8 olarak belirlenmiştir. Her müşterinin her talebi bu ortalama ve standart sapmaya göre normal dağılımdan üretilmiştir $X \sim N(1.6\ , 0.8^2)$ .

Yukarıdaki veri değerlerine sahip 10 rassal problem üretilmiştir. Sonrasında daha zor olmasını beklediğimiz 10 problem üretilmesine karar verilmiştir ve $a$ değeri 0.10'dan 0.25'e yükseltilmiştir. Yükseltilen $a$ oranı ile 10 problem daha üretilmiştir. Üretilen 20 problem için, skala ekonomik faktörünün aynı talep değerleri üzerine etkisinin incelemesi için belirlenen sef faktörü 0.70'ten 0.85 değerine yükseltilerek üretilen 20 probleme alternatif olarak 20 problem daha üretilmiştir. Aynı oranlara sahip 40 problem için CV değeri 0.5'ten 1'e yükseltilmiş ve 40 problem daha üretilmiştir. Üretilen 80 problem Tablo 5.2.1'de gösterilmiştir.

Tablo 5.2.1: Üretilen 80 problemin sahip olduğu oranlar

| Problem Sayısı | CV | a | sef |
|---|---|---|---|
| 10 | 0.5 | 0.1 | 0.70 |
| 10 | 0.5 | 0.1 | 0.85 |
| 10 | 0.5 | 0.25 | 0.70 |
| 10 | 0.5 | 0.25 | 0.85 |
| 10 | 1 | 0.1 | 0.70 |
| 10 | 1 | 0.1 | 0.85 |
| 10 | 1 | 0.25 | 0.70 |
| 10 | 1 | 0.25 | 0.85 |

Yukarıdaki oranlara sahip 80 problem üretilmiş ve sonuçlar incelenmiştir. Buradaki amaç oranların problem zorluğuna ve problem çözümüne ne derecede etki edeceğini bulmaktır.

## 5.3. Problem Çözümü ve Analizi

Bu tez kapsamındaki metot Eclipse Java programlama dili ile kodlanmıştır ve matematiksel modeller CPLEX çözücüsünde Intel 2.50 GHz i5-3210M işlemcili, 4 GB RAM kapasiteli bilgisayarda işlenmiştir. Sabitle ve optimize et metodunun bahsedilen 4 farklı uygulanışı sonucunda 320 adet sonuç elde edilmiştir. Uygulanan 4 farklı yöntem için sonuçlar ve performanslar incelenmiştir.

### 5.3.1. Çoklu Periyot Bazlı Yaklaşım

Tez kapsamında bu yöntem uygulanırken ana problem 3'er periyotluk gruplar halinde küçük problemlere bölünmüştür. Yapılan performans değerlendirmeleri sonucunda, optimal çözümlere en yakın değerler veren yöntemin 3-periyotlu bazlı

sabitle ve optimize et metodu olduğu görülmüştür. Tablo 5.3.1.1 – 5.3.1.8 arasında üretilen 8 set, 80 problem için sonuçlar gösterilmiştir.

Tüm tablolarda optimal çözüm, sezgisellerin 1. ve 2. aşamalarından elde edilen çözümler ve 2. aşama sonunda bulunan çözümün varsa optimalden, yoksa alt sınırdan yüzde sapması ve alt-sınırın optimalden yüzde sapması verilmiştir. Alt-sınır sonuçları 3. ve 4. set problemlerde bu problemlerin optimallerini bulmanın çok uzun süre gerektirmesi nedeniyle kullanılmıştır. Alt sınır değerleri 9 periyotluk problemin son iki peryodu için değişkenlerin doğrusal gevşetilip problemin çözdürülmesiyle elde edilmiştir.

Tablo 5.3.1.1: İlk problem seti için çoklu periyot bazlı yaklaşım sonuçları

| **α=0,10 CV=0,5 Sef=0,70** | **Optimal** | **Alt Sınır** | **SOE 1** | **SOE 2** | **Fark(%)** | **AOF (%)** |
|---|---|---|---|---|---|---|
| 1 | 1740,00 | 1488,16 | 1740,00 | 1740,00 | 0,00 | 14,47 |
| 2 | 1738,94 | 1580,70 | 1748,32 | 1748,32 | 0,54 | 9,10 |
| 3 | 1745,46 | 1601,31 | 1749,16 | 1749,16 | 0,21 | 8,26 |
| 4 | 1754,69 | 1587,91 | 1765,98 | 1765,98 | 0,64 | 9,50 |
| 5 | 1722,23 | 1572,03 | 1722,60 | 1722,60 | 0,02 | 8,72 |
| 6 | 1719,77 | 1470,72 | 1719,77 | 1719,77 | 0,00 | 14,48 |
| 7 | 1736,83 | 1489,72 | 1737,03 | 1737,03 | 0,01 | 14,23 |
| 8 | 1731,03 | 1490,22 | 1731,03 | 1731,03 | 0,00 | 13,91 |
| 9 | 1732,79 | 1476,00 | 1732,79 | 1732,79 | 0,00 | 14,82 |
| 10 | 1719,30 | 1378,29 | 1719,30 | 1719,30 | 0,00 | 19,83 |
| Ortalama | - | - | - | - | 0,14 | 12,73 |

Tablo 5.3.1.2: İkinci problem seti için çoklu periyot bazlı yaklaşım sonuçları

| **α=0,10 CV=0,5 Sef=0,85** | **Optimal** | **Alt Sınır** | **SOE 1** | **SOE 2** | **Fark(%)** | **AOF(%)** |
|---|---|---|---|---|---|---|
| 1 | 1759,22 | 1580,77 | 1769,53 | 1769,53 | 0,59 | 10,14 |
| 2 | 1755,31 | 1580,78 | 1764,10 | 1764,10 | 0,50 | 9,94 |
| 3 | 1762,12 | 1599,71 | 1769,26 | 1769,26 | 0,40 | 9,22 |
| 4 | 1771,18 | 1587,91 | 1781,31 | 1781,31 | 0,57 | 10,35 |
| 5 | 1736,41 | 1572,03 | 1744,58 | 1744,58 | 0,47 | 9,47 |
| 6 | 1739,00 | 1555,38 | 1739,00 | 1739,00 | 0,00 | 10,56 |
| 7 | 1750,40 | 1581,02 | 1762,41 | 1760,71 | 0,59 | 9,68 |
| 8 | 1750,25 | 1574,42 | 1760,56 | 1760,56 | 0,59 | 10,05 |
| 9 | 1742,68 | 1564,01 | 1752,39 | 1752,39 | 0,56 | 10,25 |
| 10 | 1748,83 | 1552,93 | 1748,83 | 1748,83 | 0,00 | 11,20 |
| Ortalama | - | - | - | - | 0,43 | 10,09 |

Birinci ve ikinci set problemlerinde ortalama fark tablolardan görüleceği üzere yaklaşık %0.30'dur. Algoritmanın performansı ilk iki set problemde oldukça iyi çıkmıştır.

Tablo 5.3.1.3: Üçüncü problem seti için çoklu periyot bazlı yaklaşım sonuçları

| **α =0,25 CV=0,50 Sef=0,70** | **Alt Sınır** | **SOE 1** | **SOE 2** | **Fark(%)** |
|---|---|---|---|---|
| **1** | 2438,557 | 2610,370 | 2610,370 | 7,046 |
| **2** | 2341,740 | 2481,613 | 2481,613 | 5,973 |
| **3** | 2275,223 | 2470,462 | 2470,462 | 8,581 |
| **4** | 2424,500 | 2532,249 | 2532,249 | 4,444 |
| **5** | 2279,639 | 2465,407 | 2465,407 | 8,149 |
| **6** | 2455,783 | 2608,783 | 2608,783 | 6,230 |
| **7** | 2456,619 | 2591,165 | 2591,165 | 5,477 |
| **8** | 2332,686 | 2448,893 | 2448,893 | 4,982 |
| **9** | 2205,544 | 2322,336 | 2322,336 | 5,295 |
| **10** | 2238,946 | 2276,541 | 2276,541 | 1,679 |
| Ortalama | - | - | - | 5,786 |

Tablo 5.3.1.4: Dördüncü problem seti için çoklu periyot bazlı yaklaşım sonuçları

| **α =0,25 CV=0,50 Sef=0,70** | **Alt Sınır** | **Sabitle ve Optimize Et(Aşama 1)** | **Sabitle ve Optimize Et(Aşama 2)** | **Fark(%)** |
|---|---|---|---|---|
| **1** | 2621,870 | 2754,980 | 2754,980 | 5,077 |
| **2** | 2554,106 | 2684,395 | 2684,395 | 5,101 |
| **3** | 2515,860 | 2730,200 | 2730,200 | 8,520 |
| **4** | 2756,012 | 2912,650 | 2912,650 | 5,684 |
| **5** | 2586,057 | 2729,079 | 2729,079 | 5,531 |
| **6** | 2786,811 | 2928,700 | 2928,700 | 5,091 |
| **7** | 2784,758 | 2910,190 | 2910,190 | 4,504 |
| **8** | 2635,749 | 2792,738 | 2792,738 | 5,956 |
| **9** | 2456,741 | 2608,273 | 2608,273 | 6,168 |
| **10** | 2535,821 | 2609,666 | 2609,666 | 2,912 |
| Ortalama | - | - | - | 5,454 |

Üçüncü ve dördüncü set problemlerinde modellerin optimal çözümü CPLEX çözücüsü ile makul sürede çözülememiştir. Bu problem setleri için modelin sadece 8. ve 9. periyotları gevşetilmiş değişkenler olarak, kalanı ise tamsayılı değişkenler olarak bırakıldıktan sonra model çözülmüş ve geçerli bir alt sınır bulunmuştur. Tablo 5.3.1.1 ve 5.3.1.2 incelendiğinde optimal çözüm ile belirtilen yöntem ile bulunan alt sınır değeri arasındaki farkın ortalama olarak %10 olduğu görülmektedir.

Üçüncü ve dördüncü problem setinde alt sınır ile algoritma değerleri arasında ortalama %5'lik bir fark vardır. Daha önceki problemlerde alt sınırlar optimale %10 sapma gösterdiği için yedinci ve sekizinci problem setlerinde algoritmanın bulduğu çözümlerin optimal değerlere %5'ten çok daha yakın olduğunu değerlendirmek mümkündür.

Tablo 5.3.1.5: Beşinci problem seti için çoklu periyot bazlı yaklaşım sonuçları

| α=0,10 CV=1 Sef=0,70 | Optimal | Alt Sınır | SOE 1 | SOE 2 | Fark(%) | AOF (%) |
|---|---|---|---|---|---|---|
| 1 | 1596,84 | 1402,51 | 1600,19 | 1600,19 | 0,21 | 12,17 |
| 2 | 1591,35 | 1392,57 | 1626,37 | 1626,37 | 2,15 | 12,49 |
| 3 | 1570,91 | 1366,22 | 1570,91 | 1570,91 | 0,00 | 13,03 |
| 4 | 1594,31 | 1396,21 | 1641,76 | 1641,76 | 2,89 | 12,43 |
| 5 | 1636,38 | 1422,16 | 1636,38 | 1636,38 | 0,00 | 13,09 |
| 6 | 1637,68 | 1405,71 | 1640,59 | 1640,59 | 0,18 | 14,16 |
| 7 | 1543,71 | 1371,59 | 1591,47 | 1591,47 | 3,00 | 11,15 |
| 8 | 1551,85 | 1334,53 | 1557,33 | 1557,33 | 0,35 | 14,00 |
| 9 | 1584,22 | 1427,45 | 1584,22 | 1584,22 | 0,00 | 9,90 |
| 10 | 1507,23 | 1347,83 | 1507,23 | 1507,23 | 0,00 | 10,58 |
| Ortalama | - | - | - | - | 0,88 | 12,30 |

Tablo 5.3.1.6: Altıncı problem seti için çoklu periyot bazlı yaklaşım sonuçları

| α=0,10 CV=1 Sef=0,85 | Optimal | Alt Sınır | SOE 1 | SOE 2 | Fark(%) | AOF (%) |
|---|---|---|---|---|---|---|
| 1 | 1596,84 | 1402,51 | 1600,19 | 1600,19 | 0,21 | 12,17 |
| 2 | 1591,35 | 1409,97 | 1626,37 | 1626,37 | 2,20 | 11,40 |
| 3 | 1570,91 | 1377,79 | 1570,91 | 1570,91 | 0,00 | 12,29 |
| 4 | 1594,31 | 1411,09 | 1641,76 | 1641,76 | 2,98 | 11,49 |
| 5 | 1636,38 | 1437,89 | 1636,38 | 1636,38 | 0,00 | 12,13 |
| 6 | 1637,68 | 1422,88 | 1640,59 | 1640,59 | 0,18 | 13,12 |
| 7 | 1543,71 | 1383,43 | 1591,47 | 1591,47 | 3,09 | 10,38 |
| 8 | 1551,85 | 1344,24 | 1557,33 | 1557,33 | 0,35 | 13,38 |
| 9 | 1584,22 | 1441,59 | 1584,22 | 1584,22 | 0,00 | 9,00 |
| 10 | 1507,23 | 1357,15 | 1507,23 | 1507,23 | 0,00 | 9,96 |
| Ortalama | - | - | - | - | 0,90 | 11,53 |

Beşinci ve altıncı problem seti için optimal çözümler ile sabitle & optimize et algoritmasının çoklu periyot bazlı yaklaşımı ile ulaşılan çözümler arasındaki farklar tablodan görülebileceği gibi ortalama olarak %1'in altındadır. Sonuçlar algoritmanın çoklu-periyot bazlı yaklaşımının performansının oldukça iyi olduğunu göstermektedir. Altıncı problem setinde amaç fonksiyon değerlerinin artma sebebi skala ekonomi faktörünün 0.70 değerinden 0.85 değerine yükseltilmesidir. Faktör

yükseltildiğinde problem verileri aynı kalırken, büyük araçları kullanmanın maliyeti artmaktadır. Bu artış da amaç fonksiyonundaki maliyetler toplamına yansımıştır.

Problemin optimal çözüm değeri yaklaşık 15 dakikada bulunurken, sezgisel metot ile problem yaklaşık 30 saniyede çözülebilmektedir. Bu süreler problem boyutu büyüdüğünde sezgiselin kısa sürede etkili çözümler vereceğine bir kanıt olmaktadır.

Tablo 5.3.1.7: Yedinci problem seti için çoklu periyot bazlı yaklaşım sonuçları

| **α=0,25 CV=1 Sef=0,70** | **Optimal** | **Alt Sınır** | **SOE 1** | **SOE 2** | **Fark(%)** | **AOF** |
|---|---|---|---|---|---|---|
| 1 | 1793,88 | 1689,72 | 1816,71 | 1816,71 | 1,27 | 5,81 |
| 2 | 1672,17 | 1632,05 | 1672,17 | 1672,17 | 0,00 | 2,40 |
| 3 | 1758,74 | 1678,25 | 1769,05 | 1769,05 | 0,59 | 4,58 |
| 4 | 1788,69 | 1700,16 | 1805,46 | 1805,46 | 0,94 | 4,95 |
| 5 | 1692,71 | 1579,41 | 1702,08 | 1702,08 | 0,55 | 6,69 |
| 6 | 1780,07 | 1718,63 | 1795,60 | 1789,44 | 0,53 | 3,45 |
| 7 | 1743,89 | 1640,38 | 1743,89 | 1743,89 | 0,00 | 5,94 |
| 8 | 1788,88 | 1707,47 | 1809,06 | 1809,06 | 1,13 | 4,55 |
| 9 | 1774,57 | 1734,88 | 1774,57 | 1774,57 | 0,00 | 2,24 |
| 10 | 1659,56 | 1536,61 | 1669,40 | 1669,40 | 0,59 | 7,41 |
| Ortalama | - | - | - | - | 0,56 | 4,80 |

Tablo 5.3.1.8: Sekizinci problem seti için çoklu periyot bazlı yaklaşım sonuçları

| **α=0,25 CV=1 Sef=0,85** | **Optimal** | **Alt Sınır** | **SOE 1** | **SOE 2** | **Fark(%)** | **AOF (%)** |
|---|---|---|---|---|---|---|
| 1 | 1905,01 | 1850,71 | 1955,70 | 1955,70 | 2,66 | 2,85 |
| 2 | 1821,00 | 1766,00 | 1821,00 | 1821,00 | 0,00 | 3,02 |
| 3 | 1908,56 | 1834,96 | 1908,56 | 1908,56 | 0,00 | 3,86 |
| 4 | 1927,33 | 1834,72 | 1927,33 | 1927,33 | 0,00 | 4,81 |
| 5 | 1793,48 | 1720,37 | 1829,37 | 1829,37 | 2,00 | 4,08 |
| 6 | 1880,85 | 1820,92 | 1896,38 | 1896,38 | 0,83 | 3,19 |
| 7 | 1868,11 | 1788,48 | 1938,65 | 1938,65 | 3,78 | 4,26 |
| 8 | 1869,59 | 1807,30 | 1933,70 | 1933,70 | 3,43 | 3,33 |
| 9 | 1962,29 | 1915,64 | 1962,29 | 1962,29 | 0,00 | 2,38 |
| 10 | 1759,16 | 1678,45 | 1793,62 | 1793,62 | 1,96 | 4,59 |
| Ortalama | - | - | - | - | 1,47 | 3,64 |

Yedinci ve sekizinci problem setleri içinde optimal ile algoritma arasındaki ortalama fark yaklaşık %1'dir. Bu setteki problemlerin altıncı ve yedinci setteki problemlemlere göre farkı $a$ oranının 0.10 değerinden 0.25 değerine arttırılmasıdır. $a$ değeri arttığında araç kapasiteleri ve sayısı sabitken müşterilerin bir periyottaki ortalama talep miktarı artmaktadır. Bu da problemin zorlaşmasına sebep olmaktadır.

### 5.3.2. Tek Periyot Bazlı Yaklaşım

Algoritmanın anlatımında bahsedildiği gibi tek periyot bazlı yaklaşım çözülemeyen büyük problemler için çoklu periyot bazlı yaklaşıma alternatif olarak öne sürülmüştür. Sonuçlar tablo 5.3.2.1 ile 5.3.2.8 arasında gösterilmiştir. 80 adet problemde, problem setine bağlı olarak, ortalama olarak optimale %4.5 – 16 arası yaklaşan çözümlerin elde edildiği görülmüştür.

Tablo 5.3.2.1: İlk problem seti için tek periyot bazlı yaklaşım sonuçları

| **α=0,10 CV=0,5 Sef=0,70** | **Optimal** | **Sabitle&Optimize Et(Aşama 1)** | **Sabitle&Optimize Et(Aşama 2)** | **Fark(%)** |
|---|---|---|---|---|
| 1 | 1740,00 | 1840,36 | 1821,57 | 4,48 |
| 2 | 1738,94 | 1883,11 | 1856,02 | 6,31 |
| 3 | 1745,46 | 1912,44 | 1887,61 | 7,53 |
| 4 | 1754,69 | 1805,01 | 1805,01 | 2,79 |
| 5 | 1722,23 | 1723,73 | 1723,73 | 0,09 |
| 6 | 1719,77 | 1799,65 | 1764,04 | 2,51 |
| 7 | 1736,83 | 1909,66 | 1800,04 | 3,51 |
| 8 | 1731,03 | 1913,86 | 1873,52 | 7,61 |
| 9 | 1732,79 | 2091,70 | 1912,79 | 9,41 |
| 10 | 1719,30 | 1955,82 | 1904,31 | 9,72 |
| Ortalama | - | - | - | 5,39 |

Tablo 5.3.2.2: İlk problem seti için tek periyot bazlı yaklaşım sonuçları

| **α=0,10 CV=0,50 Sef=0,85** | **Optimal** | **Sabitle&Optimize Et(Aşama 1)** | **Sabitle&Optimize Et(Aşama 2)** | **Fark(%)** |
|---|---|---|---|---|
| 1 | 1759,22 | 1875,25 | 1846,91 | 4,75 |
| 2 | 1755,31 | 1883,11 | 1856,02 | 5,43 |
| 3 | 1762,12 | 1852,14 | 1852,14 | 4,86 |
| 4 | 1771,18 | 1805,01 | 1805,01 | 1,87 |
| 5 | 1736,41 | 1796,13 | 1796,13 | 3,32 |
| 6 | 1739,00 | 1871,23 | 1802,94 | 3,55 |
| 7 | 1750,40 | 1812,73 | 1768,16 | 1,00 |
| 8 | 1750,25 | 1858,99 | 1812,74 | 3,45 |
| 9 | 1742,68 | 1891,00 | 1891,00 | 7,84 |
| 10 | 1748,83 | 1955,82 | 1904,31 | 8,16 |
| Ortalama | - | - | - | 4,42 |

Tablo 5.3.2.3: Üçüncü problem seti için tek periyot bazlı yaklaşım sonuçları

| **α =0,25 CV=0,50 Sef=0,70** | **Alt Sınır** | **Sabitle&Optimize Et(Aşama 1)** | **Sabitle&Optimize Et(Aşama 2)** | **Fark(%)** |
|---|---|---|---|---|
| **1** | 2438,56 | 2949,79 | 2696,15 | 10,56 |
| **2** | 2341,74 | 3037,16 | 2776,46 | 18,56 |
| **3** | 2275,22 | 2845,57 | 2654,68 | 16,68 |
| **4** | 2424,50 | 2875,01 | 2714,06 | 11,94 |
| **5** | 2279,64 | 2551,89 | 2486,54 | 9,08 |
| **6** | 2455,78 | 2801,07 | 2699,08 | 9,91 |
| **7** | 2456,62 | 2886,94 | 2785,07 | 13,37 |
| **8** | 2332,69 | 2754,53 | 2644,55 | 13,37 |
| **9** | 2205,54 | 2661,90 | 2501,67 | 13,43 |
| **10** | 2238,95 | 2478,57 | 2402,70 | 7,31 |
| Ortalama | - | - | - | 12,42 |

Tablo 5.3.2.4: Dördüncü problem seti için tek periyot bazlı yaklaşım sonuçları

| **α =0,25 CV=0,50 Sef=0,85** | **Alt Sınır** | **Sabitle&Optimize Et(Aşama 1)** | **Sabitle&Optimize Et(Aşama 2)** | **Fark(%)** |
|---|---|---|---|---|
| **1** | 2621,87 | 3307,34 | 3077,66 | 17,38 |
| **2** | 2554,11 | 3302,03 | 3046,03 | 19,26 |
| **3** | 2515,86 | 3251,15 | 2985,09 | 18,65 |
| **4** | 2756,01 | 3445,67 | 3114,54 | 13,01 |
| **5** | 2586,06 | 3216,00 | 3051,95 | 18,02 |
| **6** | 2786,81 | 3447,92 | 3219,59 | 15,53 |
| **7** | 2784,76 | 3456,72 | 3223,94 | 15,77 |
| **8** | 2635,75 | 3365,86 | 3145,67 | 19,35 |
| **9** | 2456,74 | 3104,85 | 2892,35 | 17,73 |
| **10** | 2535,82 | 3195,67 | 2911,52 | 14,82 |
| Ortalama | - | - | - | 16,95 |

Tablo 5.3.2.5: Beşinci problem seti için tek periyot bazlı yaklaşım sonuçları

| **α=0,10 CV=1 Sef=0,7** | **Optimal** | **Sabitle&Optimize Et(Aşama 1)** | **Sabitle&Optimize Et(Aşama 2)** | **Fark(%)** |
|---|---|---|---|---|
| 1 | 1596,84 | 1828,65 | 1790,34 | 10,81 |
| 2 | 1591,35 | 1929,71 | 1765,55 | 9,87 |
| 3 | 1570,91 | 1860,97 | 1794,23 | 12,45 |
| 4 | 1594,31 | 1963,22 | 1801,54 | 11,50 |
| 5 | 1636,38 | 1841,82 | 1826,35 | 10,40 |
| 6 | 1637,68 | 1948,82 | 1830,54 | 10,54 |
| 7 | 1543,71 | 1826,89 | 1787,18 | 13,62 |
| 8 | 1551,85 | 1862,26 | 1792,34 | 13,42 |
| 9 | 1584,22 | 1891,98 | 1785,67 | 11,28 |
| 10 | 1507,23 | 1809,36 | 1729,36 | 12,84 |
| Ortalama | - | - | - | 11,67 |

Tablo 5.3.2.6: Altıncı problem seti için tek periyot bazlı yaklaşım sonuçları

| **α=0,10 CV=1 Sef=0,85** | **Optimal** | **Sabitle&Optimize Et(Aşama 1)** | **Sabitle&Optimize Et(Aşama 2)** | **Fark(%)** |
|---|---|---|---|---|
| 1 | 1596,84 | 1942,34 | 1790,34 | 12,12 |
| 2 | 1591,35 | 1821,93 | 1765,55 | 10,95 |
| 3 | 1570,91 | 1942,23 | 1794,23 | 14,22 |
| 4 | 1594,31 | 1881,99 | 1801,54 | 13,00 |
| 5 | 1636,38 | 1902,91 | 1826,35 | 11,61 |
| 6 | 1637,68 | 1944,59 | 1830,54 | 11,78 |
| 7 | 1543,71 | 1918,76 | 1787,18 | 15,77 |
| 8 | 1551,85 | 1913,07 | 1792,34 | 15,50 |
| 9 | 1584,22 | 1899,91 | 1785,67 | 12,72 |
| 10 | 1507,23 | 1846,54 | 1729,36 | 14,74 |
| Ortalama | - | - | - | 13,24 |

Tablo 5.3.2.7: Yedinci problem seti için tek periyot bazlı yaklaşım sonuçları

| **α=0,10 CV=1 Sef=0,85** | **Optimal** | **Sabitle&Optimize Et(Aşama 1)** | **Sabitle&Optimize Et(Aşama 2)** | **Fark(%)** |
|---|---|---|---|---|
| 1 | 1793,88 | 2131,17 | 1966,79 | 9,64 |
| 2 | 1672,17 | 1979,04 | 1890,56 | 13,06 |
| 3 | 1758,74 | 2018,83 | 1925,15 | 9,46 |
| 4 | 1788,69 | 1998,75 | 1986,74 | 11,07 |
| 5 | 1692,71 | 1954,07 | 1946,34 | 14,98 |
| 6 | 1780,07 | 2019,64 | 1954,34 | 9,79 |
| 7 | 1743,89 | 2014,16 | 1960,15 | 12,40 |
| 8 | 1788,88 | 1957,94 | 1939,22 | 8,40 |
| 9 | 1774,57 | 2172,71 | 1977,38 | 11,43 |
| 10 | 1659,56 | 2079,70 | 1921,01 | 15,75 |
| Ortalama | - | - | - | 11,60 |

Tablo 5.3.2.8: Sekizinci problem seti için tek periyot bazlı yaklaşım sonuçları

| α=0,25 CV=1 Sef=0,85 | Optimal | Sabitle&Optimize Et(Aşama 1) | Sabitle&Optimize Et(Aşama 2) | Fark(%) |
|---|---|---|---|---|
| 1 | 1905,01 | 2263,86 | 2221,56 | 16,62 |
| 2 | 1821,00 | 2299,73 | 2110,77 | 15,91 |
| 3 | 1908,56 | 2314,80 | 2142,74 | 12,27 |
| 4 | 1927,33 | 2351,65 | 2300,20 | 19,35 |
| 5 | 1793,48 | 2158,12 | 2103,54 | 17,29 |
| 6 | 1880,85 | 2257,74 | 2135,74 | 13,55 |
| 7 | 1868,11 | 2184,29 | 2165,77 | 15,93 |
| 8 | 1869,59 | 2222,60 | 2191,08 | 17,20 |
| 9 | 1962,29 | 2401,70 | 2357,61 | 20,15 |
| 10 | 1759,16 | 2099,98 | 2004,37 | 13,94 |
| Ortalama | - | - | - | 16,22 |

Sekiz problem setinin ikisi hariç performans optimalle karşılaştırılmış ve ortalama itibariyle %4,4 ile %13,2 arası bir optalamadan sapma gözlenmiştir. İki problemde de alt sınırdan sapma değerleri elde edilmiştir. Bu sapmalara bakıldığında ve altsınırın optimalden sapmalrını daha önceki problemlere göre değerlendirdiğimizde optimalden sapmanın bu problemlerde %10 altında olacağını düşünmekteyiz.

### 5.3.3. İki Periyotlu Kayan Pencere Yöntemi

Üçüncü yaklaşım olan iki periyotlu kayan pencere yönteminin sonuçları Tablo 5.3.3.1 ile 5.3.3.8 arasında gösterilmiştir. Bu yöntem orta derece büyüklükteki problemlerde iyi performanslı çözüm elde edilmesi için oluşturulmuştur. Çoklu periyot bazlı probleme göre bir grupta daha az periyot çözmesi sebebiyle çözüm süresi açısından etkin bir yöntemdir. Ancak diğer yöntemlere göre çözüm süresi açısından biraz daha uzun sürmektedir. Bu nedenle boyutları çok büyük olan ve çoklu periyot bazlı yaklaşım ile makul sürede çözüm alınamayan problemler için iki periyotlu kayan pencere yöntemi denenebilir. Bu yöntem ile de problemler makul

sürede çözülemezse tek grupta daha küçük problemler çözen diğer üç yaklaşımdan biri problem üzerinde uygulanabilmektedir. Aynı zamanda sonuçlardan görüleceği gibi çoklu periyot bazlı yaklaşıma çok yakın çözüm değerleri vermektedir. Bu da çoklu periyot bazlı yaklaşım uygulanamadığında en iyi performanslı yöntemin iki periyotlu kayan pencere yöntemi olduğunu göstermektedir.

Tablo 5.3.3.1: Birinci problem seti için İPKP sonuçları

| **α=0,1 CV=0,5 Sef=0,70** | **Optimal** | **Sabitle&Optimize Et(Aşama 1)** | **Sabitle&Optimize Et(Aşama 2)** | **Fark(%)** |
|---|---|---|---|---|
| 1 | 1740,00 | 1780,07 | 1753,36 | 0,76 |
| 2 | 1738,94 | 1748,32 | 1748,32 | 0,54 |
| 3 | 1745,46 | 1761,23 | 1761,24 | 0,90 |
| 4 | 1754,69 | 1765,98 | 1765,98 | 0,64 |
| 5 | 1722,23 | 1738,14 | 1738,14 | 0,92 |
| 6 | 1719,77 | 1719,77 | 1719,77 | 0,00 |
| 7 | 1736,83 | 1737,03 | 1737,03 | 0,01 |
| 8 | 1731,03 | 1731,03 | 1731,03 | 0,00 |
| 9 | 1732,79 | 1732,79 | 1732,79 | 0,00 |
| 10 | 1719,30 | 1740,25 | 1740,25 | 1,20 |
| Ortalama | - | - | - | 0,50 |

Tablo 5.3.3.2: İkinci problem seti için İPKP sonuçları

| **α=0,1 CV=0,5 Sef=0,85** | **Optimal** | **Sabitle&Optimize Et(Aşama 1)** | **Sabitle&Optimize Et(Aşama 2)** | **Fark(%)** |
|---|---|---|---|---|
| 1 | 1759,22 | 1769,53 | 1769,53 | 0,59 |
| 2 | 1755,31 | 1780,10 | 1777,36 | 1,26 |
| 3 | 1762,12 | 1781,33 | 1781,33 | 1,09 |
| 4 | 1771,18 | 1781,31 | 1781,31 | 0,57 |
| 5 | 1736,41 | 1776,34 | 1776,34 | 2,30 |
| 6 | 1739,00 | 1751,30 | 1751,30 | 0,71 |
| 7 | 1750,40 | 1762,41 | 1760,71 | 0,59 |
| 8 | 1750,25 | 1760,56 | 1760,56 | 0,59 |
| 9 | 1742,68 | 1752,39 | 1752,39 | 0,56 |
| 10 | 1748,83 | 1748,83 | 1748,83 | 0,00 |
| Ortalama | - | - | - | 0,82 |

Tablo 5.3.3.3: Üçüncü problem seti için İPKP sonuçları

| **α=0,25 CV=0,5 Sef=0,70** | **Alt Sınır** | **Sabitle&Optimize Et(Aşama 1)** | **Sabitle&Optimize Et(Aşama 2)** | **Fark(%)** |
|---|---|---|---|---|
| 1 | 2438,56 | 2634,45 | 2634,45 | 8,03 |
| 2 | 2341,74 | 2511,61 | 2511,61 | 7,25 |
| 3 | 2275,22 | 2512,33 | 2512,33 | 10,42 |
| 4 | 2424,50 | 2601,03 | 2601,03 | 7,28 |
| 5 | 2279,64 | 2465,41 | 2465,41 | 8,15 |
| 6 | 2455,78 | 2621,79 | 2621,79 | 6,76 |
| 7 | 2456,62 | 2651,33 | 2651,33 | 7,93 |
| 8 | 2332,69 | 2495,91 | 2495,91 | 7,00 |
| 9 | 2205,54 | 2385,11 | 2385,11 | 8,14 |
| 10 | 2238,95 | 2325,38 | 2325,38 | 3,86 |
| Ortalama | - | - | - | 7,48 |

Tablo 5.3.3.4: Dördüncü problem seti için İPKP sonuçları

| **α=0,25 CV=0,5 Sef=0,85** | **Alt Sınır** | **Sabitle&Optimize Et(Aşama 1)** | **Sabitle&Optimize Et(Aşama 2)** | **Fark(%)** |
|---|---|---|---|---|
| 1 | 2621,87 | 2773,25 | 2773,25 | 5,77 |
| 2 | 2554,11 | 2734,13 | 2734,13 | 7,05 |
| 3 | 2515,86 | 2761,30 | 2761,30 | 9,76 |
| 4 | 2756,01 | 2957,63 | 2957,63 | 7,32 |
| 5 | 2586,06 | 2729,08 | 2729,08 | 5,53 |
| 6 | 2786,81 | 2938,15 | 2938,15 | 5,43 |
| 7 | 2784,76 | 2934,05 | 2934,05 | 5,36 |
| 8 | 2635,75 | 2812,76 | 2812,76 | 6,72 |
| 9 | 2456,74 | 2633,23 | 2633,23 | 7,18 |
| 10 | 2535,82 | 2643,04 | 2643,04 | 4,23 |
| Ortalama | - | - | - | 6,43 |

Tablo 5.3.3.5: Beşinci problem seti için İPKP sonuçları

| **α=0,1 CV=1 Sef=0,70** | **Optimal** | **Sabitle&Optimize Et(Aşama 1)** | **Sabitle&Optimize Et(Aşama 2)** | **Fark(%)** |
|---|---|---|---|---|
| 1 | 1596,84 | 1619,10 | 1619,10 | 1,37 |
| 2 | 1591,35 | 1602,05 | 1602,05 | 0,67 |
| 3 | 1570,91 | 1603,20 | 1603,20 | 2,01 |
| 4 | 1594,31 | 1691,23 | 1691,23 | 5,73 |
| 5 | 1636,38 | 1636,38 | 1636,38 | 0,00 |
| 6 | 1637,68 | 1664,11 | 1664,11 | 1,59 |
| 7 | 1543,71 | 1600,05 | 1600,05 | 3,52 |
| 8 | 1551,85 | 1617,22 | 1617,22 | 4,04 |
| 9 | 1584,22 | 1584,22 | 1584,22 | 0,00 |
| 10 | 1507,23 | 1507,23 | 1507,23 | 0,00 |
| Ortalama | - | - | - | 1,89 |

Tablo 5.3.3.6: Altıncı problem seti için İPKP sonuçları

| **α=0,1 CV=1 Sef=0,85** | **Optimal** | **Sabitle&Optimize Et(Aşama 1)** | **Sabitle&Optimize Et(Aşama 2)** | **Fark(%)** |
|---|---|---|---|---|
| 1 | 1596,84 | 1608,32 | 1608,32 | 0,72 |
| 2 | 1591,35 | 1646,35 | 1646,35 | 3,46 |
| 3 | 1570,91 | 1603,20 | 1603,20 | 2,06 |
| 4 | 1594,31 | 1691,23 | 1691,23 | 6,08 |
| 5 | 1636,38 | 1636,38 | 1636,38 | 0,00 |
| 6 | 1637,68 | 1664,11 | 1664,11 | 1,61 |
| 7 | 1543,71 | 1591,47 | 1591,47 | 3,09 |
| 8 | 1551,85 | 1617,23 | 1617,23 | 4,21 |
| 9 | 1584,22 | 1584,22 | 1584,22 | 0,00 |
| 10 | 1507,23 | 1507,23 | 1507,23 | 0,00 |
| Ortalama | - | - | - | 2,12 |

Tablo 5.3.3.7: Yedinci problem seti için İPKP sonuçları

| **α=0,25 CV=1 Sef=0,7** | **Optimal** | **Sabitle&Optimize Et(Aşama 1)** | **Sabitle&Optimize Et(Aşama 2)** | **Fark(%)** |
|---|---|---|---|---|
| 1 | 1793,88 | 1827,19 | 1827,19 | 1,86 |
| 2 | 1672,17 | 1711,22 | 1711,22 | 2,34 |
| 3 | 1758,74 | 1810,05 | 1810,05 | 2,92 |
| 4 | 1788,69 | 1825,55 | 1825,55 | 2,06 |
| 5 | 1692,71 | 1734,03 | 1734,03 | 2,44 |
| 6 | 1780,07 | 1805,43 | 1805,43 | 1,42 |
| 7 | 1743,89 | 1793,34 | 1793,34 | 2,84 |
| 8 | 1788,88 | 1831,67 | 1831,67 | 2,39 |
| 9 | 1774,57 | 1800,03 | 1800,03 | 1,43 |
| 10 | 1659,56 | 1702,21 | 1702,21 | 2,57 |
| Ortalama | - | - | - | 2,23 |

Tablo 5.3.3.8: Sekizinci problem seti için İPKP sonuçları

| **α=0,25 CV=1 Sef=0,85** | **Optimal** | **Sabitle&Optimize Et(Aşama 1)** | **Sabitle&Optimize Et(Aşama 2)** | **Fark(%)** |
|---|---|---|---|---|
| 1 | 1905,01 | 1985,36 | 1985,36 | 4,22 |
| 2 | 1821,00 | 1845,13 | 1845,13 | 1,33 |
| 3 | 1908,56 | 1938,22 | 1938,22 | 1,55 |
| 4 | 1927,33 | 1957,13 | 1957,13 | 1,55 |
| 5 | 1793,48 | 1848,33 | 1848,33 | 3,06 |
| 6 | 1880,85 | 1923,63 | 1923,63 | 2,27 |
| 7 | 1868,11 | 1965,23 | 1965,23 | 5,20 |
| 8 | 1869,59 | 1974,51 | 1974,51 | 5,61 |
| 9 | 1962,29 | 1962,29 | 1962,29 | 0,00 |
| 10 | 1759,16 | 1834,16 | 1834,16 | 4,26 |
| Ortalama | - | - | - | 2,91 |

### 5.3.4. Araç ve Periyot Bazlı Yaklaşım

Aynı 80 problem için algoritmanın dördüncü yaklaşımı olan araç ve periyot bazlı yöntemin sonuçları bu bölümde incelenecektir. Tablo 5.3.4.1 ile 5.3.4.8 arasında sonuçlar aşağıda gösterilmiştir. Bu yaklaşımın çoklu periyot bazlı yaklaşımına göre daha kötü performansı olduğu, beklenilebileceği gibi, görülmüştür. Ancak problem büyük boyutlara ulaştığında, çoklu periyot bazlı yaklaşım ile çözüme ulaşmak uzun zaman alabilmektedir ve bu problemler araç ve periyot bazlı yaklaşım ile daha makul sürelerde çözülebimektedir. Bu yaklaşım ile sekiz set problemin altısında optimalden yaklaşık %10 sapan sonuçlar elde edilebilmiştir.

Tablo 5.3.4.1: İlk problem seti için araç ve periyot bazlı yaklaşım sonuçları

| **α=0,10 CV=0,50 Sef=0,70** | **Optimal** | **Sabitle&Optimize Et(Aşama 1)** | **Sabitle&Optimize Et(Aşama 2)** | **Fark(%)** |
|---|---|---|---|---|
| 1 | 1740,00 | 1955,88 | 1867,03 | 6,80 |
| 2 | 1738,94 | 1962,70 | 1896,68 | 8,32 |
| 3 | 1745,46 | 1901,20 | 1876,54 | 6,99 |
| 4 | 1754,69 | 1853,76 | 1853,76 | 5,34 |
| 5 | 1722,23 | 1873,91 | 1799,23 | 4,28 |
| 6 | 1719,77 | 2049,47 | 1854,66 | 7,27 |
| 7 | 1736,83 | 1958,41 | 1848,79 | 6,06 |
| 8 | 1731,03 | 2054,05 | 1929,32 | 10,28 |
| 9 | 1732,79 | 2190,78 | 2072,85 | 16,41 |
| 10 | 1719,30 | 1954,01 | 1945,66 | 11,63 |
| Ortalama | - | - | - | 8,34 |

Tablo 5.3.4.2: İkinci problem seti için araç ve periyot bazlı yaklaşım sonuçları

| **α=0,10 CV=0,50 Sef=0,85** | **Optimal** | **Sabitle&Optimize Et(Aşama 1)** | **Sabitle&Optimize Et(Aşama 2)** | **Fark(%)** |
|---|---|---|---|---|
| 1 | 1759,22 | 1982,42 | 1932,66 | 8,97 |
| 2 | 1755,31 | 1962,70 | 1896,68 | 7,45 |
| 3 | 1762,12 | 1921,35 | 1895,68 | 7,05 |
| 4 | 1771,18 | 1894,65 | 1876,51 | 5,61 |
| 5 | 1736,41 | 1924,56 | 1900,02 | 8,61 |
| 6 | 1739,00 | 2100,45 | 2015,29 | 13,71 |
| 7 | 1750,40 | 1992,71 | 1911,72 | 8,44 |
| 8 | 1750,25 | 2113,92 | 2001,54 | 12,55 |
| 9 | 1742,68 | 2256,32 | 2113,78 | 17,56 |
| 10 | 1748,83 | 2104,55 | 1985,61 | 11,92 |
| Ortalama | - | - | - | 10,19 |

Tablo 5.3.4.3: Üçüncü problem seti için araç ve periyot bazlı yaklaşım sonuçları

| **α =0,25 CV=0,50 Sef=0,70** | **Alt Sınır** | **Sabitle&Optimize Et(Aşama 1)** | **Sabitle&Optimize Et(Aşama 2)** | **Fark(%)** |
|---|---|---|---|---|
| **1** | 2438,56 | 2682,06 | 2682,06 | 9,99 |
| **2** | 2341,74 | 2569,19 | 2569,19 | 9,71 |
| **3** | 2275,22 | 2500,46 | 2500,46 | 9,90 |
| **4** | 2424,50 | 2611,86 | 2611,86 | 7,73 |
| **5** | 2279,64 | 2542,29 | 2475,25 | 8,58 |
| **6** | 2455,78 | 2628,30 | 2619,64 | 6,67 |
| **7** | 2456,62 | 2682,84 | 2682,84 | 9,21 |
| **8** | 2332,69 | 2650,14 | 2633,13 | 12,88 |
| **9** | 2205,54 | 2345,54 | 2345,54 | 6,35 |
| **10** | 2238,95 | 2467,97 | 2457,35 | 9,75 |
| Ortalama | - | - | - | 9,08 |

Tablo 5.3.4.4: Dördüncü problem seti için araç ve periyot bazlı yaklaşım sonuçları

| **α =0,25 CV=0,50 Sef=0,85** | **Alt Sınır** | **Sabitle&Optimize Et(Aşama 1)** | **Sabitle&Optimize Et(Aşama 2)** | **Fark(%)** |
|---|---|---|---|---|
| **1** | 2621,87 | 3051,25 | 3040,88 | 15,98 |
| **2** | 2554,11 | 2988,65 | 2977,72 | 16,59 |
| **3** | 2515,86 | 2869,60 | 2869,60 | 14,06 |
| **4** | 2756,01 | 2930,14 | 2930,14 | 6,32 |
| **5** | 2586,06 | 2863,38 | 2796,35 | 8,13 |
| **6** | 2786,81 | 2997,44 | 2988,78 | 7,25 |
| **7** | 2784,76 | 3121,99 | 3087,48 | 10,87 |
| **8** | 2635,75 | 3018,11 | 3001,10 | 13,86 |
| **9** | 2456,74 | 2664,29 | 2664,29 | 8,45 |
| **10** | 2535,82 | 2744,12 | 2744,12 | 8,21 |
| Ortalama | - | - | - | 10,97 |

Tablo 5.3.4.5: Beşinci problem seti için araç ve periyot bazlı yaklaşım sonuçları

| **α=0,10 CV=1 Sef=0,70** | **Optimal** | **Sabitle&Optimize Et(Aşama 1)** | **Sabitle&Optimize Et(Aşama 2)** | **Fark(%)** |
|---|---|---|---|---|
| 1 | 1596,84 | 1764,86 | 1754,93 | 9,01 |
| 2 | 1591,35 | 1800,78 | 1755,87 | 9,37 |
| 3 | 1570,91 | 1703,05 | 1703,05 | 7,76 |
| 4 | 1594,31 | 1840,49 | 1797,22 | 11,29 |
| 5 | 1636,38 | 1985,01 | 1895,77 | 13,68 |
| 6 | 1637,68 | 1715,96 | 1715,96 | 4,56 |
| 7 | 1543,71 | 1777,39 | 1744,82 | 11,53 |
| 8 | 1551,85 | 1707,43 | 1626,24 | 4,57 |
| 9 | 1584,22 | 1662,97 | 1662,97 | 4,74 |
| 10 | 1507,23 | 1822,63 | 1753,98 | 14,07 |
| Ortalama | - | - | - | 9,06 |

Tablo 5.3.4.6: Altıncı problem seti için araç ve periyot bazlı yaklaşım sonuçları

| **α=0,10 CV=1 Sef=0,85** | **Optimal** | **Sabitle&Optimize Et(Aşama 1)** | **Sabitle&Optimize Et(Aşama 2)** | **Fark(%)** |
|---|---|---|---|---|
| 1 | 1596,84 | 1924,50 | 1900,01 | 18,99 |
| 2 | 1591,35 | 2098,57 | 2005,65 | 26,03 |
| 3 | 1570,91 | 2003,91 | 1973,83 | 25,65 |
| 4 | 1594,31 | 1993,05 | 1934,56 | 21,34 |
| 5 | 1636,38 | 2201,80 | 2042,22 | 24,80 |
| 6 | 1637,68 | 2134,43 | 2098,28 | 28,13 |
| 7 | 1543,71 | 1948,98 | 1915,91 | 24,11 |
| 8 | 1551,85 | 1885,55 | 1729,60 | 11,45 |
| 9 | 1584,22 | 2170,95 | 2142,42 | 35,23 |
| 10 | 1507,23 | 2003,45 | 1934,80 | 28,37 |
| Ortalama | - | - | - | 24,41 |

Tablo 5.3.4.7: Yedinci problem seti için araç ve periyot bazlı yaklaşım sonuçları

| **α=0,25 CV=1 Sef=0,7** | **Optimal** | **Sabitle&Optimize Et(Aşama 1)** | **Sabitle&Optimize Et(Aşama 2)** | **Fark(%)** |
|---|---|---|---|---|
| 1 | 1793,88 | 1946,91 | 1933,05 | 7,76 |
| 2 | 1672,17 | 1816,08 | 1816,08 | 8,61 |
| 3 | 1758,74 | 1964,49 | 1944,83 | 10,58 |
| 4 | 1788,69 | 1964,99 | 1964,99 | 9,86 |
| 5 | 1692,71 | 1829,14 | 1829,14 | 8,06 |
| 6 | 1780,07 | 2096,84 | 2096,84 | 17,80 |
| 7 | 1743,89 | 1896,99 | 1896,99 | 8,78 |
| 8 | 1788,88 | 1954,48 | 1954,48 | 9,26 |
| 9 | 1774,57 | 1922,43 | 1922,43 | 8,33 |
| 10 | 1659,56 | 1853,46 | 1853,46 | 11,68 |
| Ortalama | - | - | - | 10,07 |

Tablo 5.3.4.8: Sekizinci problem seti için araç ve periyot bazlı yaklaşım sonuçları

| **α=0,25 CV=1 Sef=0,85** | **Optimal** | **Sabitle&Optimize Et(Aşama 1)** | **Sabitle&Optimize Et(Aşama 2)** | **Fark(%)** |
|---|---|---|---|---|
| 1 | 1905,01 | 2100,35 | 2100,35 | 10,25 |
| 2 | 1821,00 | 2132,94 | 2132,94 | 17,13 |
| 3 | 1908,56 | 2183,45 | 2110,87 | 10,60 |
| 4 | 1927,33 | 2235,69 | 2235,69 | 16,00 |
| 5 | 1793,48 | 2134,02 | 2074,66 | 15,68 |
| 6 | 1880,85 | 2392,15 | 2392,15 | 27,18 |
| 7 | 1868,11 | 2141,92 | 2141,92 | 14,66 |
| 8 | 1869,59 | 2178,31 | 2178,31 | 16,51 |
| 9 | 1962,29 | 2135,25 | 2133,60 | 8,73 |
| 10 | 1759,16 | 2184,99 | 2184,99 | 24,21 |
| Ortalama | - | - | - | 16,10 |

Sekiz problem setinin ikisi hariç performans optimalle karşılaştırılmış ve ortalama itibariyle %8,3 ile %24,5 arası bir ortalamadan sapma gözlenmiştir. İki problemde

de alt sınırdan sapma değerleri elde edilmiştir. Bu sapmalara bakıldığında ve altsınırın optimalden sapmaları daha önceki problemlere göre değerlendirildiğinde optimalden sapmanın bu problemlerde %7-8'in altında olacağı düşünülmektedir.

### 5.3.5. Büyük Araç Öncelikli ve Periyot Bazlı Yaklaşım

Araç periyot bazlı yaklaşıma alternatif olarak sunulmuş uygulama yöntemidir. Tüm periyotlar boyunca önce büyük araçların sabitlenmesi ve daha sonra küçük araçların sabitleme işlemine geçilmesi aşamalarını içerir. Bu durumun sebebi, büyük araçların daha maliyetli olmasıdır. Önce büyük araçların kararlarının alınması, sonuçlarda araç-periyot bazlı yaklaşıma göre iyileşmeye neden olmuştur. 80 problem için sonuçlar 5.3.5.1 ile 5.3.5.8 arasındaki tüm tablolarda gösterilmiştir.

Tablo 5.3.5.1: İlk problem seti için BAÖP sonuçları

| **α=0,10 CV=0,5 Sef=0,70** | **Optimal** | **Sabitle&Optimize Et(Aşama 1)** | **Sabitle&Optimize Et(Aşama 2)** | **Fark(%)** |
|---|---|---|---|---|
| 1 | 1740,00 | 1857,85 | 1857,85 | 6,34 |
| 2 | 1738,94 | 1894,95 | 1877,86 | 7,40 |
| 3 | 1745,46 | 1833,10 | 1801,75 | 3,12 |
| 4 | 1754,69 | 1829,81 | 1794,00 | 2,19 |
| 5 | 1722,23 | 1797,07 | 1743,83 | 1,24 |
| 6 | 1719,77 | 2014,65 | 1814,53 | 5,22 |
| 7 | 1736,83 | 1898,26 | 1824,49 | 4,80 |
| 8 | 1731,03 | 1955,58 | 1878,74 | 7,86 |
| 9 | 1732,79 | 2162,22 | 2055,82 | 15,71 |
| 10 | 1719,30 | 1892,23 | 1876,92 | 8,40 |
| Ortalama | - | - | - | 6,23 |

Tablo 5.3.5.2: İkinci problem seti için BAÖP sonuçları

| **α=0,10 CV=0,5 Sef=0,85** | **Optimal** | **Sabitle&Optimize Et(Aşama 1)** | **Sabitle&Optimize Et(Aşama 2)** | **Fark(%)** |
|---|---|---|---|---|
| 1 | 1759,22 | 1968,99 | 1924,56 | 8,59 |
| 2 | 1755,31 | 1946,47 | 1896,44 | 7,44 |
| 3 | 1762,12 | 1864,16 | 1839,74 | 4,22 |
| 4 | 1771,18 | 1842,14 | 1842,14 | 3,85 |
| 5 | 1736,41 | 1827,36 | 1784,51 | 2,70 |
| 6 | 1739,00 | 2156,73 | 1941,09 | 10,41 |
| 7 | 1750,40 | 1972,30 | 1856,44 | 5,71 |
| 8 | 1750,25 | 2045,88 | 1857,07 | 5,75 |
| 9 | 1742,68 | 2375,56 | 2111,66 | 17,47 |
| 10 | 1748,83 | 2093,70 | 1901,22 | 8,02 |
| Ortalama | - | - | - | 7,42 |

Tablo 5.3.5.3: Üçüncü problem seti için BAÖP sonuçları

| **α =0,25 CV=0,50 Sef=0,70** | **Alt Sınır** | **Sabitle&Optimize Et(Aşama 1)** | **Sabitle&Optimize Et(Aşama 2)** | **Fark(%)** |
|---|---|---|---|---|
| **1** | 2438,56 | 2667,24 | 2641,98 | 8,34 |
| **2** | 2341,74 | 2556,18 | 2506,47 | 7,03 |
| **3** | 2275,22 | 2526,78 | 2492,97 | 9,57 |
| **4** | 2424,50 | 2594,85 | 2567,15 | 5,88 |
| **5** | 2279,64 | 2523,47 | 2454,86 | 7,69 |
| **6** | 2455,78 | 2691,02 | 2605,11 | 6,08 |
| **7** | 2456,62 | 2659,69 | 2634,64 | 7,25 |
| **8** | 2332,69 | 2642,13 | 2581,62 | 10,67 |
| **9** | 2205,54 | 2352,98 | 2323,93 | 5,37 |
| **10** | 2238,95 | 2519,32 | 2452,79 | 9,55 |
| Ortalama | - | - | - | 7,74 |

Tablo 5.3.5.4: Dördüncü problem seti için BAÖP sonuçları

| **α =0,25 CV=0,50 Sef=0,85** | **Alt Sınır** | **Sabitle&Optimize Et(Aşama 1)** | **Sabitle&Optimize Et(Aşama 2)** | **Fark(%)** |
|---|---|---|---|---|
| **1** | 2621,87 | 3051,09 | 2985,93 | 13,89 |
| **2** | 2554,11 | 2944,66 | 2922,73 | 14,43 |
| **3** | 2515,86 | 2942,13 | 2820,30 | 12,10 |
| **4** | 2756,01 | 2913,86 | 2853,89 | 3,55 |
| **5** | 2586,06 | 2847,22 | 2755,34 | 6,55 |
| **6** | 2786,81 | 3013,65 | 2981,92 | 7,00 |
| **7** | 2784,76 | 3142,65 | 3013,76 | 8,22 |
| **8** | 2635,75 | 3111,49 | 2986,31 | 13,30 |
| **9** | 2456,74 | 2684,17 | 2612,87 | 6,36 |
| **10** | 2535,82 | 2811,96 | 2743,94 | 8,21 |
| Ortalama | - | - | - | 9,36 |

Tablo 5.3.5.5: Beşinci problem seti için BAÖP sonuçları

| **α=0,10 CV=1 Sef=0,70** | **Optimal** | **Sabitle&Optimize Et(Aşama 1)** | **Sabitle&Optimize Et(Aşama 2)** | **Fark(%)** |
|---|---|---|---|---|
| 1 | 1596,84 | 1724,32 | 1713,31 | 6,80 |
| 2 | 1591,35 | 1777,94 | 1750,56 | 9,09 |
| 3 | 1570,91 | 1685,52 | 1654,63 | 5,06 |
| 4 | 1594,31 | 1802,50 | 1745,27 | 8,65 |
| 5 | 1636,38 | 1923,06 | 1861,68 | 12,10 |
| 6 | 1637,68 | 1721,72 | 1702,63 | 3,81 |
| 7 | 1543,71 | 1774,40 | 1731,08 | 10,82 |
| 8 | 1551,85 | 1656,64 | 1602,58 | 3,17 |
| 9 | 1584,22 | 1684,57 | 1636,72 | 3,21 |
| 10 | 1507,23 | 1813,93 | 1709,53 | 11,83 |
| Ortalama | - | - | - | 7,46 |

Tablo 5.3.5.6: Altıncı problem seti için BAÖP sonuçları

| **α=0,10 CV=1 Sef=0,85** | **Optimal** | **Sabitle&Optimize Et(Aşama 1)** | **Sabitle&Optimize Et(Aşama 2)** | **Fark(%)** |
|---|---|---|---|---|
| 1 | 1596,84 | 1893,59 | 1822,40 | 14,13 |
| 2 | 1591,35 | 2052,63 | 1966,97 | 23,60 |
| 3 | 1570,91 | 1967,78 | 1958,20 | 24,65 |
| 4 | 1594,31 | 1935,66 | 1913,12 | 20,00 |
| 5 | 1636,38 | 2111,50 | 2002,94 | 22,40 |
| 6 | 1637,68 | 2057,75 | 2019,69 | 23,33 |
| 7 | 1543,71 | 1898,19 | 1882,46 | 21,94 |
| 8 | 1551,85 | 1808,68 | 1643,98 | 5,94 |
| 9 | 1584,22 | 2114,90 | 2084,61 | 31,59 |
| 10 | 1507,23 | 1928,01 | 1842,53 | 22,25 |
| Ortalama | - | - | - | 20,98 |

Tablo 5.3.5.7: Yedinci problem seti için BAÖP sonuçları

| **α=0,10 CV=1 Sef=0,85** | **Optimal** | **Sabitle&Optimize Et(Aşama 1)** | **Sabitle&Optimize Et(Aşama 2)** | **Fark(%)** |
|---|---|---|---|---|
| 1 | 1793,88 | 1931,90 | 1878,52 | 4,72 |
| 2 | 1672,17 | 1798,19 | 1772,88 | 6,02 |
| 3 | 1758,74 | 1923,94 | 1913,09 | 8,78 |
| 4 | 1788,69 | 1932,90 | 1910,13 | 6,79 |
| 5 | 1692,71 | 1821,39 | 1818,89 | 7,45 |
| 6 | 1780,07 | 2143,90 | 2093,01 | 17,58 |
| 7 | 1743,89 | 1887,05 | 1852,47 | 6,23 |
| 8 | 1788,88 | 1907,84 | 1907,84 | 6,65 |
| 9 | 1774,57 | 1917,71 | 1893,70 | 6,71 |
| 10 | 1659,56 | 1851,64 | 1851,64 | 11,57 |
| Ortalama | - | - | - | 8,25 |

Tablo 5.3.5.8: Sekizinci problem seti için BAÖP sonuçları

| α=0,10 CV=1 Sef=0,85 | Optimal | Sabitle&Optimize Et(Aşama 1) | Sabitle&Optimize Et(Aşama 2) | Fark(%) |
|---|---|---|---|---|
| 1 | 1905,01 | 2059,41 | 2071,33 | 8,73 |
| 2 | 1821,00 | 2109,51 | 2064,90 | 13,39 |
| 3 | 1908,56 | 2127,09 | 2089,95 | 9,50 |
| 4 | 1927,33 | 2186,91 | 2167,06 | 12,44 |
| 5 | 1793,48 | 2069,13 | 2027,95 | 13,07 |
| 6 | 1880,85 | 2371,09 | 2354,97 | 25,21 |
| 7 | 1868,11 | 2093,79 | 2098,74 | 12,35 |
| 8 | 1869,59 | 2163,86 | 2140,66 | 14,50 |
| 9 | 1962,29 | 2128,20 | 2123,41 | 8,21 |
| 10 | 1759,16 | 2167,54 | 2128,78 | 21,01 |
| Ortalama | - | - | - | 13,84 |

## 5.4. Yöntemlerin Karşılaştırılması

Sabitle & optimize et metodunun yöntemleri karşılaştırılırken her problem seti için yöntemin optimalden ortalama sapması hesaplanmıştır. Elde edilen sonuçlarda dört yöntemin karşılaştırıldığı özet değerler tablo 5.4.1'de gösterilmiştir.

Tablo 5.4.1: Beş yöntemin her problem seti için ortalama sapması ve karşılaştırmalar

| Problem seti/Yöntem | ÇP (%) | TP(%) | İPKP(%) | AP(%) | BAÖP(%) |
|---|---|---|---|---|---|
| 1 | 0,14 | 5,39 | 0,5 | 8,34 | 6,23 |
| 2 | 0,43 | 4,42 | 0,82 | 10,19 | 7,42 |
| 3 | (5,78) | (12,42) | (7,48) | (9,08) | (7,74) |
| 4 | (5,45) | (16,95) | (6,43) | (10,97) | (9,36) |
| 5 | 0,88 | 11,67 | 1,89 | 9,06 | 7,46 |
| 6 | 0,9 | 13,24 | 2,12 | 24,41 | 20,98 |
| 7 | 0,56 | 11,6 | 2,23 | 10,07 | 8,25 |
| 8 | 1,47 | 16,22 | 2,91 | 16,1 | 13,84 |
| Ortalama | 0,73 | 10,42 | 3,05 | 13,03 | 10,7 |

Sonuçlardan anlaşılacağı gibi, algoritma yöntemleri arasında beklenildiği gibi en iyi sonuçlara çoklu periyot bazlı yaklaşım ile ulaşılmıştır. 80 problem için çoklu periyot bazlı yaklaşım ortalamada %1'in altında hata vermektedir. İki periyotlu kayan pencere yaklaşımı da optimalden %3,05 ortalama sapma vermiştir ki bu değer iyi bir sonuçtur. Ancak problem boyutları çok büyüdüğünde periyodu çoklu bölme seçeneği etkisiz bir seçenek olabilmektedir. Bunun sebebi bölünen problemlerin bile boyutlarının büyük olması ve yeterli sürede çözülemeyebileceğidir. Bu durumda çok büyük problemlerin çözümü için çoklu periyot bazlı yaklaşım kadar iyi olmasa da diğer yaklaşımlar uygulanabilmektedir. $a$ değeri düşük iken tek periyotlu yaklaşım büyük araç öncelikli yaklaşıma oranla daha iyi sonuçlar vermektedir. Ancak $a$ değerinin veya CV değerinin arttırılması durumunda büyük araç öncelikli yaklaşımın tek periyot bazlı yaklaşıma oranla daha iyi sonuçlar verdiği görülmüştür.

Büyük araç öncelikli yaklaşımın araç periyot bazlı yaklaşıma oranla daha başarılı sonuçlar verdiği görülmüştür. Bunun nedeni daha maliyetli olan büyük araçların hangi bölgeye gideceğinin tüm periyotlar boyunca karar verilmesi ve sabitlenmesinin ardından küçük araçların kararına geçilmesidir.

3. ve 4. problem setlerinde alt sınırdan sapma değerleri tabloda görülmektedir. Bu değerleri ve alt sınırların optimalden diğer problemlerdeki sapmaları düşünüldüğünde, bu problemlerde optimalden sapmaların %1-10 değerinin altında olması beklenmektedir.

Önemli bir performans ölçütü olan araç doluluk oranları da tüm çözümlerde elde edilmiştir. Tablo 5.4.2'de görüleceği gibi, araç doluluk oranları bakımından optimale en yakın sonuç çoklu periyot bazlı yöntem ile elde edilmiştir. Araçların kapasitelerinin düzgün kullanımı ile maliyetlerin azaltıldığı bilinmektedir, çoklu periyot bazlı yaklaşım ile bulunan araç doluluk oranları optimal çözümde bulunan araç doluluk oranlarına çok yakındır. Çoklu periyot bazlı yöntemden sonra beklenildiği gibi en iyi araç doluluk oranları iki periyotlu kayan pencere yöntemi ile elde edilmiştir. Diğer yaklaşımlarla bulunan araç doluluk oranları da ortlama olarak %75'in üzerindedir. Bu da yaklaşımların performansını bir kez daha ortaya koymuştur.

Tablo 5.4.2: Beş yöntemin her problem seti için ortalama araç doluluk oranları ve karşılaştırmalar

| **Problem Seti** | **Optimal (%)** | **ÇP(%)** | **TP(%)** | **İPKP(%)** | **AP(%)** | **BAÖP(%)** |
|---|---|---|---|---|---|---|
| 1 | 85,66 | 84,98 | 76,14 | 83,66 | 72,46 | 75,55 |
| 2 | 86,03 | 84,27 | 77,01 | 83,24 | 74,54 | 75,60 |
| 3 | 95,12 | 91,36 | 78,44 | 89,51 | 80,51 | 81,74 |
| 4 | 94,89 | 90,99 | 74,11 | 88,40 | 78,51 | 79,12 |
| 5 | 87,01 | 85,73 | 76,65 | 84,01 | 78,81 | 79,24 |
| 6 | 86,78 | 85,38 | 76,17 | 83,23 | 69,16 | 70,04 |
| 7 | 91,22 | 90,54 | 81,03 | 89,13 | 81,34 | 82,00 |
| 8 | 91,13 | 89,66 | 74,13 | 88,04 | 75,15 | 78,66 |
| Ortalama | 89,73 | 87,86 | 76,71 | 86,15 | 76,31 | 77,74 |

Tablo 5.4.3'te görüleceği gibi, çoklu periyot bazlı yaklaşım amaç fonksiyon değerleri bakımından en iyi yöntem olmasına rağmen çözüm süresi açısından diğer üç yöntemin gerisinde kalmaktadır.

Tablo 5.4.3: Beş yöntemin her problem seti için ortalama çözüm süreleri

| Problem Set | Optimal Çözüm Süresi (Sn.) | Alt-Sınır Çözüm Süresi (Sn.) | ÇP Çözüm Süresi (Sn.) | TP Çözüm Süresi (Sn.) | İPKP Çözüm Süresi (Sn.) | AP Çözüm Süresi (Sn.) | BAÖP Çözüm Süresi (Sn.) |
|---|---|---|---|---|---|---|---|
| 1 | 802 | - | 13 | 3 | 7 | 3 | 6 |
| 2 | 816 | - | 15 | 3 | 6 | 4 | 5 |
| 3 | - | 2732 | 72 | 8 | 44 | 7 | 9 |
| 4 | - | 2944 | 84 | 9 | 52 | 10 | 13 |
| 5 | 824 | - | 17 | 4 | 9 | 4 | 6 |
| 6 | 832 | - | 17 | 3 | 8 | 4 | 7 |
| 7 | 874 | - | 20 | 4 | 8 | 3 | 7 |
| 8 | 891 | - | 20 | 5 | 8 | 5 | 8 |
| Ortalama | - | - | 32,25 | 4,88 | 17,75 | 5 | 7,63 |

Tablo 5.4.3'ten görüldüğü gibi, tek periyot, araç ve periyot, önce büyük daha sonra küçük araç bazlı yaklaşımlar çözüm süresi açısından oldukça kısa sürmektedir. Çoklu periyot bazlı yaklaşımda, problemler daha önceden belirtildiği gibi diğer yöntemlere oranla daha büyük problem gruplarına ayrılmıştır. Belirtilen durumdan dolayı çözüm süresi daha uzun sürmektedir ancak amaç fonksiyon değeri açısından en iyi yöntemdir. Optimal çözümün bulunamadığı üçüncü ve dördüncü problem setinde her bir problem için alt-sınır yaklaşık 45 dakikada bulunmuştur. Sabitle optimize et metodunun çoklu periyot bazlı yöntemi ile 45 dakikalık süreç optimale %5'ten daha az yakın bir ortalama 1,5 dakika gibi kısa bir sürede bulunabilmiştir. Amaç fonksiyon değerlerinin optimale yakınlığı açısından çoklu periyot bazlı yöntem kadar verimli olmayan diğer yöntemler ile yaklaşık 10 saniyede çözüm bulunmuştur. Buradan tekrar anlaşılacağı üzere, makul sürede çözülemeyen büyük problemlere bu üç yöntemden biri uygulandığında kısa bir sürede çözüm elde edilecektir.

# 6. SONUÇ VE DEĞERLENDİRMELER

Bu tez çalışmasında tek üreticili çok müşterili bir sistemde taşıma planlanması için sabitle&optimize et metodunun uygulanışı ve farklı yöntemlerinin performansları değerlendirilmiştir. Bir bakliyat firmasında yaşanan gerçek hayat probleminden esinlenerek oluşturulan yeni problemde literatürde tam olarak karşılığına rastlanmasa da, çalışılan probleme benzer konular için literatürde gerekli araştırmalar yapılmıştır.

Yapılan araştırmalar sonucunda öncelikle problemin kompleksitesi ve zorluğu anlatılmış ve problem tanımı yapılmıştır. Oluşturulan yeni problemin iki farklı durumu için de matematiksel modeller geliştirilmiştir ve tez kapsamı boyunca ilk durum için geliştirilen sabitle&optimize et metodunun dört farklı şekli bu matematiksel model üzerinde uygulanmıştır.

Yöntemler uygulanmadan önce literatürde eşine rastlanmayan bu problem için hazır veri olmaması nedeniyle belirli oranlar tanımlanmıştır. Bu oranların kullanımı ile problem için veri üretilmiştir ve birbirinden farklı verilere sahip seksen adet problem elde edilmiştir.

Daha sonra oluşturulan seksen adet problem üzerinde algoritmanın dört farklı yöntemi denenerek toplamda üç yüz yirmi adet problem çözdürülmüştür. Uygulanan algoritmanın yöntemlerinin performansları gösterilmiş ve üzerinde yorumlar yapılmıştır. Geliştirilen tüm modellerin çözümü ve algoritma yaklaşımlarının uygulanışı CPLEX 12.4 ve Eclipse Java programları içinde gerçekleşmiştir. Algoritmanın uygulanması ile makul sürede oldukça iyi sonuçlara ulaşılmıştır.

Tez kapsamında yapılan çalışmalar genişletilmeye ve geliştirilmeye açıktır. İlerde yapılabilecek bazı çalışmalar şunları içerebilir;

İlk olarak çalışılan problemin ikinci durumu olan uzaklı bazlı model için sabitle&optimize et algoritmasının dört farklı yöntemi uygulanabilir ve sonuçlar karşılaştırılabilir. Bu model içinde sonuçlar itibariyle benzer bir performans beklentimizdir.

Algoritma için daha çok yöntem denenebilir. Örnek olarak bölge bazlı veya müşteri bazlı sabitle&optimize et metodu problem üzerinde denenebilir ve sonuçlar diğer yöntemlerle karşılaştırılabilir.

Çözülemeyen problem için optimale daha yakın alt sınır bulunması için belirli analizler yapılabilir ve geçerli eşitsizlikler eklenebilir.

Problemin taleplerin belirli rassal olduğu durumdaki versiyonu ele alınıp bir çözüm yaklaşımı geliştirilebilir.

## KAYNAKLAR


[1] Adulyasak, Y., Cordeau, J. Jans, R., 2015, "The production routing problem: A review of formulations and solution algorithms, The production routing problem: A review of formulations and solution algorithms", Computers & Operations Research, Vol. 55, pp. 141-152

[2] Boudia, M., Louly, M. A. O., Prins, C., 2007, "A reactive GRASP and path relinking for a combined production-distribution problem", Computers and operations research, Vol. 34, No. 11, pp. 3402-3419.

[3] Chandra, Pankaj, and Marshall L. Fisher. "Coordination of production and distribution planning." *European Journal of Operational Research* 72.3 (1994): 503-517.

[4] Coelho, C., Cordeou, J., Laporte, G., 2013, "Thirty Years of Inventory Routing", Vol. 48, No. 1, pp. 1-19.

[5] Coelho, C., Laporte, G., 2013, "The exact solution of several classes of inventory-routing problems", Vol. 40,No. 2, pp. 558-565.

[6] Coffman Jr, Edward G., Michael R. Garey, and David S. Johnson. "Approximation algorithms for bin packing: a survey." *Approximation algorithms for NP-hard problems*. PWS Publishing Co., 1996.

[7] Federgruen, Awi, Joern Meissner, and Michal Tzur. "Progressive interval heuristics for multi-item capacitated lot-sizing problems." *Operations Research*55.3 (2007): 490-502.

[8] Fleszar, Krzysztof, and Khalil S. Hindi. "New heuristics for one-dimensional bin-packing." *Computers & operations research* 29.7 (2002): 821-839.

[9] Gintner, Vitali, Natalia Kliewer, and Leena Suhl. "Solving large multiple-depot multiple-vehicle-type bus scheduling problems in practice." *OR Spectrum* 27.4 (2005): 507-523.

[10] Hakan Keskin, "Supply chain management" Nobel Yayınları

[11] Helber, Stefan, and Florian Sahling. "A fix-and-optimize approach for the multi-level capacitated lot sizing problem." *International Journal of Production Economics* 123.2 (2010): 247-256.

[12] Hvattum, Lars Magnus, and Arne Løkketangen. "Using scenario trees and progressive hedging for stochastic inventory routing problems." *Journal of Heuristics* 15.6 (2009): 527-557.

[13] Jaillet, Patrick, et al. "Delivery cost approximations for inventory routing problems in a rolling horizon framework." *Transportation Science* 36.3 (2002): 292-300

[14] Karmarkar, Narendra, and Richard M. Karp. "An efficient approximation scheme for the one-dimensional bin-packing problem." *Foundations of Computer Science, 1982. SFCS'08. 23rd Annual Symposium on*. IEEE, 1982.

[15] Kröger, Berthold. "Guillotineable bin packing: A genetic approach." *European Journal of Operational Research* 84.3 (1995): 645-661.

[16] Lodi, Andrea, Silvano Martello, and Daniele Vigo. "Heuristic and metaheuristic approaches for a class of two-dimensional bin packing problems." *INFORMS Journal on Computing* 11.4 (1999): 345-357.

[17] Moin, Noor Hasnah, and Said Salhi. "Inventory routing problems: a logistical overview." *Journal of the Operational Research Society* 58.9 (2007): 1185-1194.

[18] Munkres, James. "Algorithms for the assignment and transportation problems." *Journal of the Society for Industrial and Applied Mathematics* 5.1 (1957): 32-38.

[19] Pochet, Yves, and Laurence A. Wolsey. Production planning by mixed integer programming. Springer Science & Business Media, 2006.

[20] Ronald H. Ballou, "Business logistics/ supply chain management" Pearson Education International 5th Edition

[21] Samet, Dov, Yair Tauman, and Israel Zang. "An application of the Aumann-Shapley prices for cost allocation in transportation problems." *Mathematics of Operations Research* 9.1 (1984): 25-42.

[22] Scholl, Armin, Robert Klein, and Christian Jürgens. "BISON: A fast hybrid procedure for exactly solving the one-dimensional bin packing problem."Computers & Operations Research 24.7 (1997): 627-645.

[23] Solyalı, O., Süral, H., 2011, 2011, "A branch-and-cut algorithm using a strong formulation and a priori tour-based heuristic for an inventory routing problem", Transportation Science, Vol. 45, No. 3, pp. 335-345.

[24] "Bin Packing Problems, 2003" erişim adresi: http://www.wikipedia.org, Erişim tarihi: 6 Haziran 2014.

# EKLER

## Bölge Bazlı Model Cplex Kodu

```
{int} M=...;

 {int} B=...;
 {int} BolgeMusterileri[B]=...;

 assert
 forall(b in B,m in BolgeMusterileri[b])m in M;
 range T=1..9;



 {int} K=...;
 {int} V=...;

// float S[V] = [0.25,0.25,0.25,0.25,0.25,0.25];
 float d[M][K][T];
float Cv[V][B]=...;
 range e=1..9*1*9;
 float arrayd[e]=...;

 int w=3;
 int Cap[V]=[16,16,16,25,25,25];
 int array=1;


 dvar float+ x[M][K][T][V][T];
 dvar float+ a[M][K][T][V][T];
 dvar boolean b[M][K][T][V][T];

 dvar float+ y[V][B][T];
 dvar float+ a1[V][B][T];
 dvar boolean b1[V][B][T];

 dvar float+ L[V][T];
 dvar float+ R[M][K][T];







  range unrelax=1..6;
  range relaxation=7..9;

execute INITIALIZE {

  for ( var j in thisOplModel.M){
     for ( var i in thisOplModel.K){
```

```
      for( var T in thisOplModel.T){

        d[j][i][T] = arrayd[array];
        array++;
        }
      }
    }


  }


 minimize
sum(v in V,t1 in T,b in B)(Cv[v][b]*y[v][b][t1])+
sum(j in M,i in K,t1 in T)(R[j][i][t1]*(100/16)*4) + sum(v in V,tp
in T,t1 in T,j in M,i in K )(x[j][i][tp][v][t1]*(10));


 subject to {

  forall(j in M,i in K,t in T,v in V)
     x[j][i][t][v][t]==a[j][i][t][v][t]+b[j][i][t][v][t];

    forall(j in M,i in K,t in T,v in V, tp in unrelax)
      a[j][i][tp][v][t]==0;
      forall(j in M,i in K,t in T,v in V, tp in relaxation)
      b[j][i][tp][v][t]==0;

      forall(b in B,t in T,v in V)
     y[v][b][t]==a1[v][b][t]+b1[v][b][t];


    forall(b in B,t in unrelax,v in V)
      a1[v][b][t]==0;
      forall(b in B,t in relaxation,v in V)
      b1[v][b][t]==0;

  forall(j in M,i in K,t1 in T)
    sum(v in V,tp in T : tp<=t1)
      x[j][i][tp][v][t1]<=1;

  forall(j in M,i in K,t1 in T)
    sum(v in V,tp in T : tp<= t1-w  && t1-w>=1)
x[j][i][tp][v][t1]==0;

 forall(v in V, tp in T)
   L[v][tp] - sum(j in M,i in K,t1 in T :
t1>=tp)(x[j][i][tp][v][t1]*d[j][i][t1]) == 0;

 forall(j in M, i in K, t1 in T)
   R[j][i][t1] + sum(v in V,tp in T :
tp<=t1)(x[j][i][tp][v][t1]*d[j][i][t1])== d[j][i][t1] ;

   forall(v in V,t1 in T)
      L[v][t1]-sum(b in B)y[v][b][t1]*Cap[v]<=0 ;

 forall(b in B,v in V,tp in T)
```

```
   sum(t1 in T,i in K,j in BolgeMusterileri[b] :
t1>=tp)x[j][i][tp][v][t1] <= y[v][b][tp]*999;

 forall(v in V,t1 in T)
   sum(b in B) y[v][b][t1]<=1;

  }
```

**Sabitle & Optimize Et Metodu Java Kodu**

```
public class Tezmodel4 {
	//İndisler
	public static int M=9;
	public static int K=1;
	public static int V=6;
	public static int B=3;
	public static int T=9;
	public static int w=3;
	public static boolean a=false;
	public static int step=0;



	//Model Parameters

	public static double [][][][][] amatrix = new double
[M][K][T][V][T];
	public static double[][][] bmatrix = new double [V][B][T];
	public static double [][][][][] a2matrix = new double
[M][K][T][V][T];
	public static double[][][] b2matrix = new double [V][B][T];
	public static double [][][] ymatrix = new double [V][B][T];
	public static double[][] Lmatrix = new double [V][T];
	public static double[][] L2matrix = new double [V][T];
	public static double[][][] Rmatrix=new double[M][K][T];
	public static double[][] DimRmatrix = new double [K][T];
	public static double[][] DimR2matrix = new double[K][T];
	public static double[][][] R2matrix=new double[M][K][T];


	public static double [][] Cv = new double[V][B];
	public static double [] S = new double[V];
	public static double [][][] d = new double[M][K][T];
	public static double[] arrayd = new double[M*K*T];
	public static double[] Cap = new double [V];
	public static double[] objectivefunction = new double [200];
	public static double[] objective = new double[200];
	public static double amacterim1=0;
	public static double amacterim2=0;
	public static double amacterim3=0;
```

```
	public static IloNumVar[][][][][][] x = new IloNumVar[M][K][T][V][T];
	public static IloNumVar [][][][][][] fx = new
IloNumVar[M][K][T][V][T];
	public static IloIntVar[][][][][][] Ix = new IloIntVar[M][K][T][V][T];

	public static IloNumVar [][][][] y = new IloNumVar[V][B][T];
	public static IloIntVar [][][][] Iy = new IloIntVar[V][B][T];
	public static IloNumVar [][][][] fy = new IloNumVar[V][B][T];

	public static IloNumVar [][] L = new IloNumVar[V][T];
	public static IloNumVar [][][][] R = new IloNumVar[M][K][T];









	// THE MODEL'S OBJECTİVE FUNCTİON AND CONSTRAİNTS ARE BELOW...

	public static void getVariables () {
		try {
			IloCplex cplex=new IloCplex();
			// For making L decision variable float
				for(int v=0; v<V; v++){
					for(int t=0; t<T; t++){
						L[v][t]=cplex.numVar(0,
Double.MAX_VALUE);
					}
				}

				//Same for the R decision variable
				for(int m=0; m<M; m++){
					for(int k=0; k<K; k++){
						for(int t=0; t<T; t++){


	R[m][k][t]=cplex.numVar(0, Double.MAX_VALUE);

						}
					}
				}

				//Defining x decision variable type as
boolean
				for(int j=0; j<M; j++){
					for(int i=0; i<K; i++ ){
						for(int t=0; t<T; t++) {
							for(int v=0; v<V; v++)
{
```

```
                                                                for(int t1=0;
t1<T; t1++) {

        Ix[j][i][t][v][t1] = cplex.boolVar();
                                                                }
                                                        }
                                                }

                                        }

                                }
                                //Defining x decision variable type as
boolean
                                for(int j=0; j<M; j++){
                                        for(int i=0; i<K; i++ ){
                                                for(int t=0; t<T; t++) {
                                                        for(int v=0; v<V; v++)
{
                                                                for(int t1=0;
t1<T; t1++) {

        fx[j][i][t][v][t1] = cplex.numVar(0,Double.MAX_VALUE);
                                                                }
                                                        }
                                                }

                                        }

                                }
                                //Defining x decision variable type as
boolean
                                for(int j=0; j<M; j++){
                                        for(int i=0; i<K; i++ ){
                                                for(int t=0; t<T; t++) {
                                                        for(int v=0; v<V; v++)
{
                                                                for(int t1=0;
t1<T; t1++) {

        x[j][i][t][v][t1] = cplex.numVar(0,Double.MAX_VALUE);

                                                                }
                                                        }
                                                }

                                        }

                                }

                                //Same for the y decision variable
                                for(int v=0; v<V; v++) {
                                        for(int b=0; b<B; b++) {
```

```
                                        for(int t=0; t<T; t++) {

Iy[v][b][t]=cplex.boolVar();

                                        }
                                        }
                                  }
                            //Same for the y decision variable
                            for(int v=0; v<V; v++) {
                                  for(int b=0; b<B; b++) {
                                        for(int t=0; t<T; t++) {

fy[v][b][t]=cplex.numVar(0,Double.MAX_VALUE);

                                        }
                                        }
                                  }

                            for(int v=0; v<V; v++) {
                                  for(int b=0; b<B; b++) {
                                        for(int t=0; t<T; t++) {

y[v][b][t]=cplex.numVar(0,Double.MAX_VALUE);

                                        }
                                        }
                                  }




      } catch (IloException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
      }
}
```

```
public static void getData() {
	int array=0;
//	for(int v=0; v<V;v++) {
//		for(int b=0; b<B; b++) {
//			double a=Math.random();
//			Cv[v][b]=(1-a)*20 + (a)*60;
//
//		}
//	}
		for(int v=0; v<3; v++) {
			S[v]=10;
		}
			for(int v=3; v<6; v++) {
				S[v]=10;
			}



		Example excel=new Example();
		excel.getExcelValues();

//		for(int a=0; a<M*K*T; a++) {
//			double sayi=Math.random();
//
//			arrayd[a]=(1-sayi)*0 + sayi*10;
//		}
		int a11=0;
		for(int j=0; j<M; j++) {
			for(int i=0; i<K; i++) {
				for(int t=0; t<T; t++) {
					d[j][i][t]=arrayd[array];
					System.out.println(arrayd[array]);
					array=array+1;
					if(a11==0) {

					a11++;
					}

				}
			}
		}
}
```

```
public static void FixandOptimize (int fp, int lp, int fix1, int fix2, int
fix3, int fix4) {
	try {

		System.out.println(" ");
		System.out.println(" ");

	System.out.println("**********************************************
*");
		System.out.println(" ");
		System.out.println("STEP : "+(step+1));
		System.out.println(" ");

	System.out.println("**********************************************
*");
		System.out.println( " ");
		IloCplex cplex = new IloCplex(); //IloCplex sınıfının
komutlarını javada cplex komutuyla kullanmak için



		//objective
				IloLinearNumExpr obj = cplex.linearNumExpr();
				for(int v=0; v<V; v++){
					for(int b=0; b<B; b++) {
```

```
                                        for(int t=0; t<T; t++) {
                                    obj.addTerm(Cv[v][b], y[v][b][t]);
// this means--> sum(Cv[v][b]*y[v][b][t])--> add this to objective line

                                        }
                                    }
                              }

                                    for(int j=0; j<M; j++){
                                        for(int i=0; i<K; i++){
                                            for(int t=0; t<T; t++){

      obj.addTerm(((100/16)*4), R[j][i][t]); // now our objective is
sum(C[v][b]*y[v][b][t]) + sum(10*R[j][i][t);


                                            }
                                        }
                                    }


                                    for(int j=0; j<M; j++) {
                                        for(int i=0; i<K; i++) {
                                            for(int tp=0; tp<T; tp++) {
                                                for(int v=0; v<V; v++)
{
                                                    for(int t=0;
t<T; t++) {
                                                        if(t>=tp){
                                    obj.addTerm(S[v], x[j][i][tp][v][t]); //
objective function completed according to cplex model.

                                                        }
                                                    }
                                                }
                                            }
                                        }
                                    }

                                    cplex.addMinimize(obj); // minimize obj,
obj is sum of the 3 terms that we added above..




                              //constraints

                                    if(step>=1) {

                                        if(fix4 !=0) {
                                            for(int j=0; j<M; j++){
                                                for(int i=0; i<K; i++
){
```

```
                                                                    for(int t=fix3;
t<=fix4; t++) {
                                                                          for(int
v=0; v<V; v++) {
       for(int t1=0; t1<T; t1++) {

       IloLinearNumExpr constraints = cplex.linearNumExpr();

       constraints.addTerm(1, x[j][i][t][v][t1]);

       cplex.addEq(constraints, amatrix[j][i][t][v][t1]);
                                                                                  } }
} } }

                                                  for(int v=0; v<V; v++) {
                                                        for(int b=0; b<B; b++)
{
                                                                    for(int t=fix3;
t<=fix4; t++) {

       IloLinearNumExpr constraints = cplex.linearNumExpr();

       constraints.addTerm(1, y[v][b][t]);

       cplex.addEq(constraints, bmatrix[v][b][t]);

                                                                    } } }


//
//
                                                  }


                                               if(fix2 !=0) {
                                               for(int j=0; j<M; j++){
                                                     for(int i=0; i<K; i++ ){
                                                           for(int t=fix1;
t<=fix2; t++) {
                                                                    for(int v=0;
v<V; v++) {
                                                                          for(int
t1=0; t1<T; t1++) {

       IloLinearNumExpr constraints = cplex.linearNumExpr();

       constraints.addTerm(1, x[j][i][t][v][t1]);

       cplex.addEq(constraints, amatrix[j][i][t][v][t1]);
```

```
                                                                        }
                                                                }
                                                        }

                                                }

                                        }

                                        for(int v=0; v<V; v++) {
                                                for(int b=0; b<B; b++) {
                                                        for(int t=fix1;
t<=fix2; t++) {

                                                                IloLinearNumExpr
constraints = cplex.linearNumExpr();

        constraints.addTerm(1, y[v][b][t]);

        cplex.addEq(constraints, bmatrix[v][b][t]);

                                                        }
                                                        }
                                                }









                                        }
                                }



                                for(int j=0; j<M; j++){
                                        for(int i=0; i<K; i++ ){
                                                for(int t=0; t<T; t++) {
                                                        for(int v=0; v<V; v++)
{
                                                                for(int t1=0;
t1<T; t1++) {


        IloLinearNumExpr constraints = cplex.linearNumExpr();

        constraints.addTerm(1, x[j][i][t][v][t1]);

        cplex.addEq(constraints,
cplex.sum(Ix[j][i][t][v][t1],fx[j][i][t][v][t1]));
                                                                }
                                                        }
                                                }

                                        }
```

```
                                }
                                for(int j=0; j<M; j++){
                                      for(int i=0; i<K; i++ ){
                                            for(int t=0; t<T; t++) {
                                                  for(int v=0; v<V; v++)
{
                                                        for(int t1=0;
t1<T; t1++) {
                                                              if(t>=fp
&& t<=lp) {

      IloLinearNumExpr constraints = cplex.linearNumExpr();

      constraints.addTerm(1, fx[j][i][t][v][t1]);

      cplex.addEq(constraints, 0);
                                                              } else {


      IloLinearNumExpr constraints = cplex.linearNumExpr();

      constraints.addTerm(1, Ix[j][i][t][v][t1]);

      cplex.addEq(constraints, 0);
                                                              }
                                                              }
                                                  }
                                            }

                                      }

                                }


                                for(int v=0; v<V; v++) {
                                      for(int b=0; b<B; b++) {
                                            for(int t=0; t<T; t++) {

                                                  IloLinearNumExpr
constraints = cplex.linearNumExpr();
                                                  constraints.addTerm(1,
y[v][b][t]);

      cplex.addEq(constraints, cplex.sum(Iy[v][b][t],fy[v][b][t]));

                                            }
                                            }
                                      }


                                for(int v=0; v<V; v++) {
                                      for(int b=0; b<B; b++) {
```

```
                                        for(int t=0; t<T; t++) {
                                        if(t>=fp && t<=lp) {
                                              IloLinearNumExpr
constraints0 = cplex.linearNumExpr();
                                              constraints0.addTerm(1,
fy[v][b][t]);
      cplex.addEq(constraints0, 0);

                                        } else {
                                              IloLinearNumExpr
constraints0 = cplex.linearNumExpr();
                                              constraints0.addTerm(1,
Iy[v][b][t]);
      cplex.addEq(constraints0, 0);

                                        }
                                        }
                                        }
                                  }




                              //constraint 1
                              for(int j=0; j<M; j++) {
                                    for(int i=0; i<K; i++) {
                                          for(int t=0;t<T; t++){
                              IloLinearNumExpr constraints1 =
cplex.linearNumExpr();
                                    for(int v=0; v<V; v++){
                                          for(int tp=0; tp<T; tp++ ){
                                                if(tp<=t) {
      constraints1.addTerm(1, x[j][i][tp][v][t]);
                                                }
                                          }
                                    }
      cplex.addLe(constraints1, 1);
                                          }
                                    }
                              }

                              //constraint 2
                              for(int j=0; j<M; j++) {
                                    for(int i=0; i<K; i++) {
                                          for(int t=0;t<T; t++){
```

```
                                    IloLinearNumExpr constraints =
cplex.linearNumExpr();
                                          for(int v=0; v<V; v++){
                                          for(int tp=0; tp<T; tp++ ){
                                                if(tp<=t-w && t-w>=1) {

      constraints.addTerm(1, x[j][i][tp][v][t]);

                                                }
                                          }
                                    }

      cplex.addEq(constraints, 0);
                                          }
                                    }
                              }

                              //constraint 3
                              for(int v=0; v<V; v++){
                                    for(int tp=0; tp<T; tp++ ){
                                          IloLinearNumExpr constraints3
= cplex.linearNumExpr();
                                          constraints3.addTerm(1,
L[v][tp]);
                                          for(int j=0; j<M; j++){
                                                for(int i=0; i<K; i++)
{
                                                      for(int t=0;
t<T; t++) {
                                                            if(tp<=t){

      constraints3.addTerm(-d[j][i][t], x[j][i][tp][v][t]);
                                                            }
                                                }
                                          }
                                    }

      cplex.addEq(constraints3, 0);
                                    }
                                    }


                              //constraint 4
                              for(int j=0; j<M; j++) {
                                    for(int i=0; i<K; i++) {
                                          for(int t=0;t<T; t++){
                              IloLinearNumExpr constraints4 =
cplex.linearNumExpr();
                              constraints4.addTerm(1, R[j][i][t]);
                                    for(int v=0; v<V; v++){
                                          for(int tp=0; tp<T; tp++ ){
                                                if(tp<=t) {

      constraints4.addTerm(+d[j][i][t], x[j][i][tp][v][t]);
```

```
                                                }
                                          }
                                    }

      cplex.addEq(constraints4, d[j][i][t]);

                                          }
                                    }
                              }



                              //constraint 5
                              for(int v=0; v<V; v++) {
                                    for(int t=0; t<T; t++) {
                                          IloLinearNumExpr constraints5
= cplex.linearNumExpr();
                                          constraints5.addTerm(1,
L[v][t]);
                                          for(int b=0; b<B; b++) {

                                                constraints5.addTerm(-
Cap[v], y[v][b][t]);
                                          }
                                          cplex.addLe(constraints5, 0);

                                    }
                              }

                              //constraint 6
      /*1.bölge için */ for(int v=0; v<V; v++) {
                                    for(int tp=0; tp<T; tp++){
                                          IloLinearNumExpr
constraints61 = cplex.linearNumExpr();
                                          constraints61.addTerm(-999,
y[v][0][tp]);
                                          for(int t=0; t<T; t++){
                                                for(int i=0; i<K; i++){
                                                      for(int j=0;

j<3;j++){
                                                            if(tp<=t)

{

      constraints61.addTerm(1, x[j][i][tp][v][t]);

                                                            }
                                                            }
                                                }
                                          }

      cplex.addLe(constraints61, 0);
                                    }
                              }
```

```
	//constraint 6
/*2.bölge için */ for(int v=0; v<V; v++) {
			for(int tp=0; tp<T; tp++){
				IloLinearNumExpr constraints62 =
cplex.linearNumExpr();
				constraints62.addTerm(-999, y[v][1][tp]);
				for(int t=0; t<T; t++){
					for(int i=0; i<K; i++){
						for(int j=3; j<6;j++){
							if(tp<=t) {


	constraints62.addTerm(1, x[j][i][tp][v][t]);


							}
							}
					}
				}
							cplex.addLe(constraints62, 0);
			}
		}
/*3.bölge için */ for(int v=0; v<V; v++) {
	for(int tp=0; tp<T; tp++){
		IloLinearNumExpr constraints63 = cplex.linearNumExpr();
		constraints63.addTerm(-999, y[v][2][tp]);
		for(int t=0; t<T; t++){
			for(int i=0; i<K; i++){
				for(int j=6; j<9;j++){
					if(tp<=t) {
					constraints63.addTerm(1,
x[j][i][tp][v][t]);

					}
					}
			}
		}
				cplex.addLe(constraints63, 0);
	}
}

	//constraint 7
		for(int v=0; v<V; v++) {
			for(int t=0; t<T; t++) {
				IloLinearNumExpr constraints7 =
cplex.linearNumExpr();
				for(int b=0; b<B; b++) {
					constraints7.addTerm(1, y[v][b][t]);
				}
					cplex.addLe(constraints7, 1);
			}
		}

		if(cplex.solve()) {
```

```
                    a=true;
                    step++;


                    amacterim1=0;
                    amacterim2=0;
                    amacterim3=0;

                    System.out.println(" ");

      System.out.println("*********************************************"
);
                    System.out.println(" ");
                    System.out.println("OBJECTİVE FUNCTİON VALUE :
"+cplex.getObjValue());
                    System.out.println(" ");

      System.out.println("*********************************************"
);
                    System.out.println(" ");
                    if(step>=1) {
                    System.out.println(" ");
                    System.out.println(" ");
                    System.out.println("FİXLENMESİ İSTENEN DEĞİŞKENLER
FİXLENDİ, MODEL YENİDEN ÇÖZÜLÜYOR..");
                    System.out.println(" ");
                    System.out.println(" ");
                    }

                    if(a=true) {




                          for(int j=0; j<M; j++){
                                for(int i=0; i<K; i++ ){
                                      for(int t=fp; t<=lp; t++) {

                                            for(int v=0; v<V; v++) {
                                                  for(int t1=0; t1<T;
t1++) {


      amatrix[j][i][t][v][t1]=cplex.getValue(x[j][i][t][v][t1]);
                                                        } } } } }



                          if(step == 1) {
                                for(int j=0; j<M; j++){
                                      for(int i=0; i<K; i++ ){
                                            for(int t=0; t<=T; t++) {
```

```
                                                        for(int v=0; v<V; v++)
{
                                                              for(int t1=0;
t1<T; t1++) {
                                                                    if(t <=
t1)

      a2matrix[j][i][t][v][t1]=cplex.getValue(x[j][i][t][v][t1]);
                                                        } } } } }



                                    for(int v=0; v<V; v++) {
                                          for(int t=0; t<T; t++) {
                                                Lmatrix[v][t] =
cplex.getValue(L[v][t]);
                                          } }


                                          for(int i=0; i<K; i++) {
                                                for(int t=0; t<T; t++) {
                                                      for(int j=0; j<M; j++){
                                                      Rmatrix[j][i][t] =
cplex.getValue(R[j][i][t]);
                                                      DimRmatrix[i][t] =
DimRmatrix[i][t] + Rmatrix[j][i][t];
                                                } } }



                                    for(int v=0; v<V; v++) {
                                          for(int b=0; b<B; b++) {
                                                for(int t=0; t<T; t++) {


      b2matrix[v][b][t]=cplex.getValue(y[v][b][t]);
                                                } } }


                              }



                              for(int v=0; v<V; v++) {
                                    for(int b=0; b<B; b++) {
                                          for(int t=fp; t<=lp; t++) {


      bmatrix[v][b][t]=cplex.getValue(y[v][b][t]);
                                          } } }
```

```
                        if(step>=0) {

      objectivefunction[step]=cplex.getObjValue();

      objective[step]=Math.rint(objectivefunction[step])*10000 / 10000;
                              for(int i=0; i<K;i++){
                                    for(int t=0; t<T; t++) {
                                          DimR2matrix[i][t]=0;

                                    }
                              }



                              for(int v=0; v<V; v++) {
                                    for(int b=0; b<B; b++) {
                                          for(int t=0; t<T; t++) {

                                                ymatrix[v][b][t]=0;
                                                ymatrix[v][b][t] =
cplex.getValue(y[v][b][t]);

                                          } } }


                              for(int i=0; i<K; i++) {
                                    for(int t=0; t<T; t++) {
                                          for(int j=0; j<M; j++){

                                          R2matrix[j][i][t] =
cplex.getValue(R[j][i][t]);
                                          DimR2matrix[i][t] =
DimR2matrix[i][t] + R2matrix[j][i][t];
                                    } } }

                              for(int v=0; v<V; v++) {

                                          for(int t=0; t<T; t++) {

                                                L2matrix[v][t]=0;
                                                L2matrix[v][t] =
cplex.getValue(L[v][t]);

                                          } }



                        }

            a=false;
                  }


                  for(int v=0; v<V; v++) {
                        for(int t=0;t<T; t++) {
```

```
                    if(cplex.getValue(L[v][t]) != 0) {
                          System.out.println("L["+(v+1)+"]["+(t+1)+"]==
"+cplex.getValue(L[v][t])+";");
                          }

                          }}
                    for(int v=0; v<V; v++) {
                          for(int b=0; b<B; b++) {

                                for(int t=0;t<T; t++) {


                                      if(cplex.getValue(y[v][b][t])>0) {
                                            amacterim1=amacterim1 +
Cv[v][b]*cplex.getValue(y[v][b][t]);

      System.out.println("y["+(v+1)+"]["+(b+1)+"]["+(t+1)+"]== "
+cplex.getValue(y[v][b][t])+";");

                                      }
                                }
                          }
                    }



                    System.out.println(" ");
                    System.out.println(" ");
                    for(int j=0; j<M; j++){
                          for(int i=0; i<K; i++ ){
                                for(int t=0; t<T; t++) {

                                      for(int v=0; v<V; v++) {
                                            for(int t1=0; t1<T; t1++) {

                                                  if(t<=t1 &&
cplex.getValue(x[j][i][t][v][t1])>0 ){

      amacterim3=amacterim3+10*cplex.getValue(x[j][i][t][v][t1]);


      System.out.println("X["+(j+1)+"]"+"["+(i+1)+"]["+(t+1)+"]["+(v+1)+"]
["+(t1+1)+"]== "+cplex.getValue(x[j][i][t][v][t1])+";");
                                                  }

                                            }
                                      }
                                }

                          }

                    }

                    for(int j=0; j<M; j++){
```

```
                        for(int i=0; i<K; i++ ){
                              for(int t=0; t<T; t++) {
                                    amacterim2=amacterim2 +
(100/16)*4*cplex.getValue(R[j][i][t]);
                              if(cplex.getValue(R[j][i][t])>0) {


      System.out.println("R["+(j+1)+"]["+(i+1)+"]["+(t+1)+"]==
"+cplex.getValue(R[j][i][t])+";");
                              }

                              }}}



            } else {
                  System.out.println("Model cant solved!");
            }
                  System.out.println("AMACTERİM1 = "+amacterim1);
                  System.out.println("AMACTERİM2 = "+amacterim2);
                  System.out.println("AMACTERİM3 = "+amacterim3);
                  Example excel = new Example();
//                excel.WriteExcelValues();

      } catch (IloException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
      }




}


public static void FixandOptimize2 (int fv, int lv, int whichperiod, int
fix1v, int fix2v, int fixperiod, int adim,
            int fixlemevarmi,int fv2, int lv2, int whichperiod2) {
      try {

            System.out.println(" ");
            System.out.println(" ");

      System.out.println("**********************************************
*");
            System.out.println(" ");
            System.out.println("STEP : "+(step+1));
            System.out.println(" ");

      System.out.println("**********************************************
*");
            System.out.println( " ");
            IloCplex cplex = new IloCplex(); //IloCplex sınıfının
komutlarını javada cplex komutuyla kullanmak için
```

```
		//objective
				IloLinearNumExpr obj = cplex.linearNumExpr();
				for(int v=0; v<V; v++){
					for(int b=0; b<B; b++) {
						for(int t=0; t<T; t++) {
					obj.addTerm(Cv[v][b], y[v][b][t]);
// this means--> sum(Cv[v][b]*y[v][b][t])--> add this to objective line

					}
				}
			}

				for(int j=0; j<M; j++){
					for(int i=0; i<K; i++){
						for(int t=0; t<T; t++){

	obj.addTerm(((100/16)*4), R[j][i][t]); // now our objective is
sum(C[v][b]*y[v][b][t]) + sum(10*R[j][i][t);


						}
					}
				}


				for(int j=0; j<M; j++) {
					for(int i=0; i<K; i++) {
						for(int tp=0; tp<T; tp++) {
							for(int v=0; v<V; v++)
{
								for(int t=0;
t<T; t++) {
									if(t>=tp){
				obj.addTerm(S[v], x[j][i][tp][v][t]); //
objective function completed according to cplex model.

									}
								}
							}
						}
					}
				}

				cplex.addMinimize(obj); // minimize obj,
obj is sum of the 3 terms that we added above..



			//constraints
```

```
                                    if(step>=1) {
                                          if(lv2>0) {
                                                for(int j=0; j<M; j++){
                                                      for(int i=0; i<K; i++
){
                                                            for(int t=0;
t<T; t++) {
                                                                  for(int
v=0; v<V; v++) {
      for(int t1=0; t1<T; t1++) {
      if(t!=whichperiod2) {
      IloLinearNumExpr constraints = cplex.linearNumExpr();
      constraints.addTerm(1, x[j][i][t][v][t1]);
      cplex.addEq(constraints, amatrix[j][i][t][v][t1]);
      } else {
                                    if(v>=fv2 && v<=lv2) {

                                    } else {
                                          IloLinearNumExpr constraints =
cplex.linearNumExpr();
                                          constraints.addTerm(1,
x[j][i][whichperiod2][v][t1]);
                                          cplex.addEq(constraints,
amatrix[j][i][whichperiod2][v][t1]);
                                    }


                                                                  }
                                                                        } }
} } }
                                                      for(int v=0; v<V; v++) {
                                                            for(int b=0; b<B; b++)
{
                                                                  for(int t=0;
t<T; t++) {
      if(t!=whichperiod2) {
```

```
        IloLinearNumExpr constraints = cplex.linearNumExpr();
        constraints.addTerm(1, y[v][b][t]);
        cplex.addEq(constraints, bmatrix[v][b][t]);
                                                                    } else {
        if(v>=fv2 && v<=lv2) {

                                                                        }
else {
        IloLinearNumExpr constraints = cplex.linearNumExpr();
        constraints.addTerm(1, y[v][b][whichperiod2]);
        cplex.addEq(constraints, bmatrix[v][b][whichperiod2]);
                                                                        }
                                                                    }
                                                            } } }

                                            }


                                            if(fixlemevarmi != 0) {
                                                  for(int j=0; j<M; j++){
                                                        for(int i=0; i<K; i++
){
                                                                    for(int
v=fix1v; v<=fix2v; v++) {
        for(int t1=0; t1<T; t1++) {

        IloLinearNumExpr constraints = cplex.linearNumExpr();
        constraints.addTerm(1, x[j][i][fixperiod][v][t1]);
        cplex.addEq(constraints, amatrix[j][i][fixperiod][v][t1]);
                                                                        } }
} }
                                                  for(int v=fix1v; v<=fix2v;
v++) {
                                                        for(int b=0; b<B; b++)
{
```

```
        IloLinearNumExpr constraints = cplex.linearNumExpr();
        constraints.addTerm(1, y[v][b][fixperiod]);
        cplex.addEq(constraints, bmatrix[v][b][fixperiod]);
                                                        } } }


                                    if(adim>=1) {
                                        for(int j=0; j<M; j++){
                                            for(int i=0; i<K; i++
){
                                                for(int t=0;
t<(adim); t++){
                                                    for(int
v=0; v<V; v++) {
        for(int t1=0; t1<T; t1++) {

        IloLinearNumExpr constraints = cplex.linearNumExpr();
        constraints.addTerm(1, x[j][i][t][v][t1]);
        cplex.addEq(constraints, amatrix[j][i][t][v][t1]);
                                                                } }
} }}
                                    }
                                    if(adim>=1) {
                                        for(int v=0; v<V; v++) {
                                            for(int b=0; b<B; b++)
{
                                                for(int t=0;
t<(adim); t++) {

        IloLinearNumExpr constraints = cplex.linearNumExpr();
        constraints.addTerm(1, y[v][b][t]);
        cplex.addEq(constraints, bmatrix[v][b][t]);
                                                        }
                                                        } }
                                    }

                                }
```

```
                                    for(int j=0; j<M; j++){
                                          for(int i=0; i<K; i++ ){
                                                for(int t=0; t<T; t++) {
                                                      for(int v=0; v<V; v++)
{
                                                            for(int t1=0;
t1<T; t1++) {


      IloLinearNumExpr constraints = cplex.linearNumExpr();

      constraints.addTerm(1, x[j][i][t][v][t1]);

      cplex.addEq(constraints,
cplex.sum(Ix[j][i][t][v][t1],fx[j][i][t][v][t1]));
                                                            }
                                                      }
                                                }

                                          }

                                    }
                                    for(int j=0; j<M; j++){
                                          for(int i=0; i<K; i++ ){

                                                      for(int v=0; v<V; v++)
{
                                                            for(int t1=0;
t1<T; t1++) {
                                                                  if(v>=fv
&& v<=lv) {


      IloLinearNumExpr constraints = cplex.linearNumExpr();

      constraints.addTerm(1, fx[j][i][whichperiod][v][t1]);

      cplex.addEq(constraints, 0);
                                                                  } else {


      IloLinearNumExpr constraints = cplex.linearNumExpr();

      constraints.addTerm(1, Ix[j][i][whichperiod][v][t1]);

      cplex.addEq(constraints, 0);
                                                                  }
                                                                  }
                                                      }
                                                }

                                          }
```

```
                              for(int v=0; v<V; v++) {
                                    for(int b=0; b<B; b++) {
                                          for(int t=0; t<T; t++) {

                                                IloLinearNumExpr
constraints = cplex.linearNumExpr();
                                                constraints.addTerm(1,
y[v][b][t]);

      cplex.addEq(constraints, cplex.sum(Iy[v][b][t],fy[v][b][t]));

                                                }
                                                }
                                    }


                              for(int v=0; v<V; v++) {
                                    for(int b=0; b<B; b++) {

                                          if(v>=fv && v<=lv) {

                                                IloLinearNumExpr
constraints0 = cplex.linearNumExpr();
                                                constraints0.addTerm(1,
fy[v][b][whichperiod]);

      cplex.addEq(constraints0, 0);

                                          } else {
                                                IloLinearNumExpr
constraints0 = cplex.linearNumExpr();
                                                constraints0.addTerm(1,
Iy[v][b][whichperiod]);

      cplex.addEq(constraints0, 0);


                                          }

                                          }
                                    }




                              //constraint 1
                              for(int j=0; j<M; j++) {
                                    for(int i=0; i<K; i++) {
                                          for(int t=0;t<T; t++){
```

```
                                    IloLinearNumExpr constraints1 =
cplex.linearNumExpr();
                                          for(int v=0; v<V; v++){
                                                for(int tp=0; tp<T; tp++ ){
                                                      if(tp<=t) {

      constraints1.addTerm(1, x[j][i][tp][v][t]);

                                                      }
                                                }
                                          }

      cplex.addLe(constraints1, 1);
                                                }
                                          }
                                    }

                                    //constraint 2
                                    for(int j=0; j<M; j++) {
                                          for(int i=0; i<K; i++) {
                                                for(int t=0;t<T; t++){
                                    IloLinearNumExpr constraints =
cplex.linearNumExpr();
                                          for(int v=0; v<V; v++){
                                          for(int tp=0; tp<T; tp++ ){
                                                      if(tp<=t-w && t-w>=1) {

      constraints.addTerm(1, x[j][i][tp][v][t]);

                                                      }
                                                }
                                          }

      cplex.addEq(constraints, 0);
                                                }
                                          }
                                    }

                                    //constraint 3
                                    for(int v=0; v<V; v++){
                                          for(int tp=0; tp<T; tp++ ){
                                                IloLinearNumExpr constraints3
= cplex.linearNumExpr();
                                                constraints3.addTerm(1,
L[v][tp]);
                                                for(int j=0; j<M; j++){
                                                      for(int i=0; i<K; i++)
{
                                                            for(int t=0;
t<T; t++) {
                                                                  if(tp<=t){


      constraints3.addTerm(-d[j][i][t], x[j][i][tp][v][t]);
                                                                  }
```

```
                                                        }
                                                }
                                        }

        cplex.addEq(constraints3, 0);
                                }
                                }


                                //constraint 4
                                for(int j=0; j<M; j++) {
                                        for(int i=0; i<K; i++) {
                                                for(int t=0;t<T; t++){
                                IloLinearNumExpr constraints4 =
cplex.linearNumExpr();
                                constraints4.addTerm(1, R[j][i][t]);
                                        for(int v=0; v<V; v++){
                                                for(int tp=0; tp<T; tp++ ){
                                                        if(tp<=t) {

        constraints4.addTerm(+d[j][i][t], x[j][i][tp][v][t]);

                                                        }
                                                }
                                        }

        cplex.addEq(constraints4, d[j][i][t]);

                                                }
                                        }
                                }



                                //constraint 5
                                for(int v=0; v<V; v++) {
                                        for(int t=0; t<T; t++) {
                                                IloLinearNumExpr constraints5
= cplex.linearNumExpr();
                                                constraints5.addTerm(1,
L[v][t]);
                                                for(int b=0; b<B; b++) {


                                                        constraints5.addTerm(-
Cap[v], y[v][b][t]);
                                                }
                                                cplex.addLe(constraints5, 0);

                                        }
                                }

                                //constraint 6
        /*1.bölge için */ for(int v=0; v<V; v++) {
                                        for(int tp=0; tp<T; tp++){
```

```
                                                IloLinearNumExpr
constraints61 = cplex.linearNumExpr();
                                                constraints61.addTerm(-999,
y[v][0][tp]);
                                                for(int t=0; t<T; t++){
                                                      for(int i=0; i<K; i++){
                                                            for(int j=0;
j<3;j++){
                                                                  if(tp<=t)
{
      constraints61.addTerm(1, x[j][i][tp][v][t]);
                                                                  }
                                                                  }
                                                      }
                                                }
      cplex.addLe(constraints61, 0);
                                          }
                                    }

      //constraint 6
/*2.bölge için */ for(int v=0; v<V; v++) {
                  for(int tp=0; tp<T; tp++){
                        IloLinearNumExpr constraints62 =
cplex.linearNumExpr();
                        constraints62.addTerm(-999, y[v][1][tp]);
                        for(int t=0; t<T; t++){
                              for(int i=0; i<K; i++){
                                    for(int j=3; j<6;j++){
                                          if(tp<=t) {

      constraints62.addTerm(1, x[j][i][tp][v][t]);

                                          }
                                          }
                              }
                        }
                                          cplex.addLe(constraints62, 0);
                  }
            }
/*3.bölge için */ for(int v=0; v<V; v++) {
      for(int tp=0; tp<T; tp++){
            IloLinearNumExpr constraints63 = cplex.linearNumExpr();
            constraints63.addTerm(-999, y[v][2][tp]);
            for(int t=0; t<T; t++){
                  for(int i=0; i<K; i++){
                        for(int j=6; j<9;j++){
                              if(tp<=t) {
                              constraints63.addTerm(1,
x[j][i][tp][v][t]);
```

```
                                        }
                                        }
                                }
                        }
                                cplex.addLe(constraints63, 0);
        }
}

        //constraint 7
                for(int v=0; v<V; v++) {
                        for(int t=0; t<T; t++) {
                                IloLinearNumExpr constraints7 =
cplex.linearNumExpr();
                                for(int b=0; b<B; b++) {
                                        constraints7.addTerm(1, y[v][b][t]);
                                }
                                        cplex.addLe(constraints7, 1);
                        }
                }

                if(cplex.solve()) {

                        a=true;



                        amacterim1=0;
                        amacterim2=0;
                        amacterim3=0;

                        System.out.println(" ");

        System.out.println("***********************************************"
);
                        System.out.println(" ");
                        System.out.println("OBJECTİVE FUNCTİON VALUE :
"+cplex.getObjValue());
                        System.out.println(" ");

        System.out.println("***********************************************"
);
                        System.out.println(" ");
                        if(step>=1) {
                        System.out.println(" ");
                        System.out.println(" ");
                        System.out.println("FİXLENMESİ İSTENEN DEĞİŞKENLER
FİXLENDİ, MODEL YENİDEN ÇÖZÜLÜYOR..");
                        System.out.println(" ");
                        System.out.println(" ");
                        }

                        if(a=true) {


                                step++;
```

```
                        for(int j=0; j<M; j++){
                              for(int i=0; i<K; i++ ){


                                          for(int v=fv; v<=lv; v++) {
                                                for(int t1=0; t1<T;
t1++) {


      amatrix[j][i][whichperiod][v][t1]=cplex.getValue(x[j][i][whichperiod
][v][t1]);

                                                      } } } }

                        for(int v=fv; v<=lv; v++) {
                              for(int b=0; b<B; b++) {



      bmatrix[v][b][whichperiod]=cplex.getValue(y[v][b][whichperiod]);
//                                          for(int t=0; t<=whichperiod;
t++) {
//
      System.out.println("b["+v+"]["+b+"]["+t+"]= "+bmatrix[v][b][t]);
//                                          }
                                    } }



                        if(step == 1) {
//                            for(int j=0; j<M; j++){
//                                  for(int i=0; i<K; i++ ){
//                                        for(int t=0; t<=T; t++) {
//
//                                              for(int v=0; v<V; v++)
{
//                                                    for(int t1=0;
t1<T; t1++) {
//                                                          if(t <=
t1)
//
//
      a2matrix[j][i][t][v][t1]=cplex.getValue(x[j][i][t][v][t1]);
//                                                      } } } } }
//


                              for(int v=0; v<V; v++) {
                                    for(int t=0; t<T; t++) {
                                          Lmatrix[v][t] =
cplex.getValue(L[v][t]);
                              } }
```

```
                                        for(int i=0; i<K; i++) {
                                              for(int t=0; t<T; t++) {
                                                    for(int j=0; j<M; j++){
                                                    Rmatrix[j][i][t] =
cplex.getValue(R[j][i][t]);
                                                    DimRmatrix[i][t] =
DimRmatrix[i][t] + Rmatrix[j][i][t];
                                              } } }


                                  for(int v=0; v<V; v++) {
                                        for(int b=0; b<B; b++) {
                                              for(int t=0; t<T; t++) {

      b2matrix[v][b][t]=cplex.getValue(y[v][b][t]);
                                              } } }


                              }









                              if(step>=0) {

      objectivefunction[step]=cplex.getObjValue();

      objective[step]=Math.rint(objectivefunction[step])*10000 / 10000;
                                  for(int i=0; i<K;i++){
                                        for(int t=0; t<T; t++) {
                                              DimR2matrix[i][t]=0;

                                        }
                                  }


                                  for(int v=0; v<V; v++) {
                                        for(int b=0; b<B; b++) {
                                              for(int t=0; t<T; t++) {

                                                    ymatrix[v][b][t]=0;
                                                    ymatrix[v][b][t] =
cplex.getValue(y[v][b][t]);

                                              } } }
```

```
                                for(int i=0; i<K; i++) {
                                      for(int t=0; t<T; t++) {
                                            for(int j=0; j<M; j++){

                                            R2matrix[j][i][t] =
cplex.getValue(R[j][i][t]);
                                            DimR2matrix[i][t] =
DimR2matrix[i][t] + R2matrix[j][i][t];
                                      } } }

                                for(int v=0; v<V; v++) {

                                            for(int t=0; t<T; t++) {

                                                  L2matrix[v][t]=0;
                                                  L2matrix[v][t] =
cplex.getValue(L[v][t]);

                                            } }



                           }

                a=false;
                      }


                     for(int v=0; v<V; v++) {
                           for(int t=0;t<T; t++) {
                     if(cplex.getValue(L[v][t]) != 0) {
                           System.out.println("L["+(v+1)+"]["+(t+1)+"]==
"+cplex.getValue(L[v][t])+";");
                           }

                           }}
                     for(int v=0; v<V; v++) {
                           for(int b=0; b<B; b++) {

                                 for(int t=0;t<T; t++) {


                                      if(cplex.getValue(y[v][b][t])>0) {
                                            amacterim1=amacterim1 +
Cv[v][b]*cplex.getValue(y[v][b][t]);

      System.out.println("y["+(v+1)+"]["+(b+1)+"]["+(t+1)+"]== "
+cplex.getValue(y[v][b][t])+";");

                                      }
                                 }
                           }
                     }
```

```
                    System.out.println(" ");
                    System.out.println(" ");
                    for(int j=0; j<M; j++){
                          for(int i=0; i<K; i++ ){
                                for(int t=0; t<T; t++) {

                                      for(int v=0; v<V; v++) {
                                            for(int t1=0; t1<T; t1++) {

                                                  if(t<=t1 &&
cplex.getValue(x[j][i][t][v][t1])>0 ){

      amacterim3=amacterim3+10*cplex.getValue(x[j][i][t][v][t1]);


      System.out.println("X["+(j+1)+"]"+"["+(i+1)+"]["+(t+1)+"]["+(v+1)+"]
["+(t1+1)+"]== "+cplex.getValue(x[j][i][t][v][t1])+";");
                                                  }

                                            }
                                      }
                                }

                          }

                    }

                    for(int j=0; j<M; j++){
                          for(int i=0; i<K; i++ ){
                                for(int t=0; t<T; t++) {
                                      amacterim2=amacterim2 +
(100/16)*4*cplex.getValue(R[j][i][t]);
                                if(cplex.getValue(R[j][i][t])>0) {


      System.out.println("R["+(j+1)+"]["+(i+1)+"]["+(t+1)+"]==
"+cplex.getValue(R[j][i][t])+";");
                                }

                                }}}



              } else {
                    System.out.println("Model cant solved!");
              }
                    System.out.println("AMACTERİM1 = "+amacterim1);
                    System.out.println("AMACTERİM2 = "+amacterim2);
                    System.out.println("AMACTERİM3 = "+amacterim3);
                    Example excel = new Example();
//                  excel.WriteExcelValues();
```

```
} catch (IloException e) {
       // TODO Auto-generated catch block
       e.printStackTrace();
} } }
```

# ÖZGEÇMİŞ

## Kişisel Bilgiler


| | | |
|---|---|---|
| Soyadı, Adı | : | TOKGÖZ, Serkan |
| Uyruğu | : | T.C. |
| Doğum tarihi ve yeri | : | 20.11.1991 Bolu |
| Medeni Hali | : | Bekar |
| Telefon | : | 0530 607 74 63 |
| e-mail | : | Serkantokgoz09@gmail.com |


## Eğitim

| Derece | Eğitim Birimi | Mezuniyet Tarihi |
|---|---|---|
| Lisans | TOBB ETU Endüstri Mühendisliği | 2013 |

## İş Deneyimi

| Yıl | Yer | Görev |
|---|---|---|
| 2013-2015 | TOBB ETU | Burslu Yüksek Lisans Öğrencisi |

## Yabancı Dil

İngilizce

Almanca

## Yayınlar

[1] ERTOĞRAL K., TOKGÖZ S., A FIX AND OPTIMIZE HEURISTIC FOR TRANSPORTATION PLANNING IN A SINGLE PRODUCER MULTI BUYER SYSTEM, 13th International Logistics and Supply Chain Congress (LM-SCM 2015), 22-23 October 2015